PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-167852

(43)Date of publication of application : 25.06.1996

(51)Int.Cl. H03M 7/40
G06F 5/00

(21)Application number : 06-308662

(22)Date of filing : 13.12.1994

(71)Applicant : FUJITSU LTD

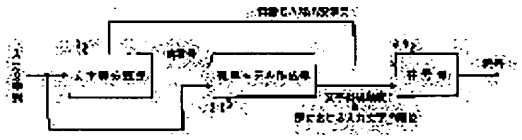
(72)Inventor : SATO NOBUKO
OKADA YOSHIYUKI
YOSHIDA SHIGERU

(54) METHOD AND DEVICE FOR COMPRESSING DATA

(57)Abstract:

PURPOSE: To attain a sufficient compressiblity factor even when the size of an object to be compressed is small by sorting characters into plural groups each of which consists of characters having the same statistic property and calculating the appearance probability of each group.

CONSTITUTION: A character string is inputted to a character group sorting part 10 and characters included in the character string are sorted into plural hierarchical groups in each character group having the same statistic property. Then a probability model preparing part 20 calculates the appearance probability of respective groups and the appearance probability of input characters in plural groups. An encoding part 30 encodes each input character based upon the calculated intra-group character appearance probability. Even when file size to be compressed is not sufficiently large size for the construction of a probability model, a high compressibility factor can be obtained without previously storing individual character appearance frequency. In the case of sorting characters into plural groups and calculating the appearance probability of respective groups, it is preferable to previously fix and apply the sorts of constitutional elements in respective groups and the appearance probability of respective groups.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

DERWENT-ACC-NO: 1996-352544
DERWENT-WEEK: 199635

TITLE: Data e.g. image data, character data compression method of computer - involves input character coding step to provide code for input characters based on appearance probability calculated at character appearance probability calculation step

PATENT-ASSIGNEE: FUJITSU LTD [FUIT]

PRIORITY-DATA: 1994JP-0308662 (December 13, 1994)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
JP 08167852 A	June 25, 1996	N/A	018	H03M 007/40

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO	APPL-DATE
JP 08167852A	N/A	1994JP-0308662	December 13, 1994

INT-CL (IPC): G06F005/00, H03M007/40

ABSTRACTED-PUB-NO: JP 08167852A

BASIC-ABSTRACT:
The method uses a group composition step (S1) which classifies the input characters into a number of hierarchical groups. A group appearance probability calculation step (S2) calculates the appearance probability of each group. A character appearance probability calculation step (S3) calculates the appearance probability of the input characters in these groups. An input coding step provides a code for the input character based on the appearance probability calculated at the character appearance probability calculation step.

ADVANTAGE - Obtains optimum compression rate without holding each character appearance frequency. Avoids increase of compression rate when data for compression is less. Performs compression based on frequency of appearance of characters.

CHOSEN-DRAWING: Dwg.1/16
TITLE-TERMS: DATA IMAGE DATA CHARACTER DATA COMPRESS METHOD COMPUTER INPUT CHARACTER CODE STEP CODE INPUT CHARACTER BASED APPEAR PROBABILITY CALCULATE CHARACTER APPEAR PROBABILITY CALCULATE STEP

DERWENT-CLASS: T01 U21
EPI-CODES: T01-D02; U21-A05A2A
SECONDARY-ACC-NO:
Non-CPI Secondary Accession Numbers: N1996-297413

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] In the data compression approach which compresses data by encoding to a variable-length sign with the code length [alphabetic character / which was inputted] according to the appearance probability The group configuration step which has the same statistical property for the alphabetic character which may be inputted mutually and which is classified into two or more hierarchical groups for every alphabetic character, respectively, The group appearance probability count step which calculates the appearance probability of each of said group, The data compression approach characterized by having the alphabetic character appearance probability count step in a group which calculates the appearance probability of the input-statement character in said two or more groups, and the input-statement character coding step which encodes an input-statement character based on the appearance probability calculated at said alphabetic character appearance probability count step in a group.
[Claim 2] The data compression approach according to claim 1 characterized by fixing beforehand and giving a classification of the component of said group at said group configuration step.
[Claim 3] The data compression approach according to claim 1 characterized by fixing beforehand and giving the appearance probability of said group at said group appearance probability count step.
[Claim 4] The data compression approach according to claim 1 characterized by re-calculating the appearance probability of this group dynamically according to the input of said alphabetic character at said group appearance probability count step while setting initial value as the appearance probability of said group beforehand.
[Claim 5] The data compression approach according to claim 1 characterized by calculating the appearance probability of said group at said group appearance probability count step according to the conditional group appearance probability on condition of each group to which two or more last characters belong appearing.
[Claim 6] The data compression approach according to claim 1 characterized by constituting said two or more hierarchical groups from the 1st group constituted in a Takaide present probability alphabetic character, and the 2nd group constituted in a low appearance probability alphabetic character at said group configuration step.
[Claim 7] In the data compression equipment which compresses data by encoding to a variable-length sign with the code length [alphabetic character / which was inputted] according to the appearance probability The group configuration section which has the same statistical property for the alphabetic character which may be inputted mutually and which is classified into two or more hierarchical groups for every alphabetic character, respectively, The group appearance probability count section which calculates the appearance probability of each of said group, and the alphabetic character appearance probability count section in a group which calculates the appearance probability of the input-statement character in said two or more groups, Data compression equipment characterized by having the input-statement character coding section which encodes an input-statement character based on the appearance probability calculated in said alphabetic character appearance probability count section in a group.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application] The amount of data dealt with is also increasing in connection with various data, such as a character code and image data, coming to be treated by computer in recent years. By excluding and compressing the redundant part in data, such a lot of data can reduce storage capacity, or can transmit it now early.
[0002] On the other hand, since it is necessary to restore them in case compressed data are referred to and used, an access rate falls compared with data before compressing. Then, the old data compression is used only at the time of the backup of data with mainly rare referring to a part, or a communication link.
[0003] However, in order to use only for [LSI] compression, the restoration rate of compressed data becomes short and it is possible [it] to perform compression and restoration also to the usual data and the data accessed similarly with recent years.
[0004] then, the data size compressed with the data size unit to access in order to restore for every compressed data unit, when it compresses -- being comparable (5 K bytes or less, about 1-2 K bytes) -- to carry out is desired.

[0005]
[Description of the Prior Art] The universal coding method is proposed as a data compression method applicable to the data (a character code, image data, etc.) of various classes. Here, although this invention is not limited to compression of a character code but it can apply to various data, below based on information theory, an alphabetic character (alphabet), a call, and data make 1 word unit of data call an arbitration WORD rope ****** thing a character string.
[0006] As a typical method, there is an algebraic-sign-ized method in a universal coding method. Like Huffman coding often used conventionally, every one character of this method outputs matching and below decimal point of a binary number to one point of coding as a sign scatteringly.
[0007] Here, the principle of the formation of a multiple-value algebraic sign is explained with reference to drawing 2 . First, it is a basic idea to express a character string with an algebraic sign using the section of the real number which is between the real numbers 0 and 1 (0 [1).
[0008] here -- section [-- 0 and 1 are adopted for outputting below decimal point of a binary number as a sign. Moreover, the reason which is "[" and following"" above is because below decimal point of 0 and 1 becomes the same and it becomes impossible to distinguish 0 and 1 in [0, 1], and is because it becomes impossible to use 0 as a value in (0, 1).
[0009] Drawing 2 (A) shows the frequency of occurrence of each alphabetic character, when it is assumed that four characters, a, b, c, and d, appear. (4) described with the alphabetic character a, b, c, and d down side of an axis of abscissa, (2), (1), and (3) show the frequency-of-occurrence ranking of each alphabetic character among drawing 2 (A).
[0010] Based on the frequency of occurrence of each alphabetic character shown in drawing 2 (A), drawing 2 (B) showed the accumulation frequency-of-occurrence probability for every alphabetic character in order of the frequency of occurrence. That is, the train described as cf0 on the axis of abscissa shows the accumulation frequency-of-occurrence probability of the alphabetic character c in four characters, c, b, d, and a, among drawing 2 (B). Similarly, the train described as cf1 on the axis of abscissa shows the accumulation frequency-of-occurrence probability of the alphabetic character b in three characters, b, d, and a. Similarly, the column

described as cf2 on the axis of abscissa shows the accumulation frequency-of-occurrence probability of the alphabetic character d in two characters, d and a. Similarly, the column described as cf3 on the axis of abscissa shows the accumulation frequency-of-occurrence probability of the alphabetic character a.

[0011] And drawing 2 (C) showed how to perform algebraic-sign-ization from the accumulation frequency-of-occurrence probability shown in drawing 2 (B). That is, section width of face equivalent to the accumulation frequency-of-occurrence probability (part where the slash was attached in the train described as cf0 in drawing 2 (B)) of the alphabetic character c is adopted as the corresponding section 10 in the phase which inputted the alphabetic character c.

[0012] Next, the section 11 which re-divides the section 10 corresponding to the 1st character by the accumulation frequency-of-occurrence probability of each alphabetic character as the corresponding section 11, and is obtained in the phase where the 2nd alphabetic character a was inputted is adopted.

[0013] And the section 12 which re-divides the section 11 corresponding to the 2nd character by the accumulation frequency-of-occurrence probability of each alphabetic character as the corresponding section 12, and is obtained in the phase where the 3rd alphabetic character d was inputted is adopted.

[0014] Thus, a character string acd is encoded as an arbitration value (any value between the upper limit of the section 12, and a lower limit) of the section 12. Here, the lower limit of each section is called for by formula (1-1) - (1-2).

The lower limit of the new partial section = accumulation of the lower limit + present partial section width-of-face x attention alphabetic character of the present partial section Probability ... (1-1)

New partial section width of face Probability of a = present partial section width-of-face x attention alphabetic character ... (1-2)

In addition, what is necessary is just to investigate, being contained at which section that the symbolic language divided into the probability of each alphabetic character, or re-dividing serially, in order to restore a symbolic language.

[0015] Thus, by algebraic-sign-ization, although it encodes in the section, in the process to decode, there is no need that the section is actually given and one certain number in the section should just be specified. What is necessary is just to choose what can be expressed with the shortest possible number of bits in the number within the section as a concrete symbolic language.

[0016] That is, since it says that section width of face becomes large so that the frequency of occurrence is high, the number below decimal point decreases, so that section width of face is large, and it can express with the short number of bits. Although it is explanation of the example which fixed each symbol frequency of occurrence, as shown below, the above can change the frequency of occurrence (probability model) serially, and can also perform it dynamically.

The accumulation probability of an attention alphabetic character = accumulation [of the count of an appearance of an alphabetic character with the frequency of occurrence lower than an attention alphabetic character]/ Input string length ... (2-1)

The probability of an attention alphabetic character = the count of an appearance / input string length of an attention alphabetic character ... (2-2)

The configuration of the equipment which performs dynamic algebraic-sign-ization which re-calculates the frequency of occurrence here at every alphabetic character input is shown in drawing 3 . This equipment consists of the probability probability-model (symbol frequency of occurrence) creation section 20 which creates the accumulation frequency of occurrence of the order of the frequency of occurrence for every alphabetic character like drawing 2 (B), and the algebraic-sign section 40 which performs algebraic-sign-ization from an accumulation frequency-of-occurrence probability like drawing 2 (C) while creating the frequency of occurrence of an inputted alphabetic character like drawing 2 (A). And the probability-model creation section 20 has the dictionary and counter which are not illustrated.

[0017] Next, the flow of drawing 4 explains actuation of the algebraic-sign-ized equipment of drawing 3 . First, let upper limit =1, lower limit =0, and section width-of-face =1.0 be the initial value of algebraic-sign-izing at step 401. At this time, the dictionary of the probability-model creation section 20 holds a symbol and frequency-of-occurrence ranking, and a counter holds each symbol frequency of occurrence. Moreover, as initialization, the dictionary of the number of symbols (256 the number of alphabetic characters which can consider an appearance: when it is 1 byte) is prepared, the counter which counts the frequency of occurrence for every alphabetic

character is prepared, and it initializes to "1." And the algebraic-sign section 40 holds the ranking and the accumulation frequency of occurrence of each symbol.
[0018] a single-character (referred to as k) input is carried out from an input string (step 402) -- every -- frequency-of-occurrence ranking is chosen from a dictionary, and the section is calculated and algebraic-sign-ized in the algebraic-sign section 40 using this number and the accumulation frequency of occurrence (step 403). That is, it asks for the upper limit and lower limit of the section of an input-statement character based on formula (1-1) - (1-2) and formula (2-1) - (2-2), and the any value of the section is outputted as a sign.
[0019] then, a counter -- the frequency of occurrence of an input-statement character -- "1" -- it increases (step 404). "1" -- the alphabetic character which increased -- following -- the order of frequency -- a dictionary -- rearranging (step 405) -- the accumulation frequency of occurrence is updated (step 406). In addition, step 405 and step 406 may be processed conversely.
[0020] Repeat activation of the actuation from the above step 402 to step 406 is carried out. Old explanation is an example in the case of algebraic-sign-izing based on the appearance probability for every single character. Furthermore, in order to raise compressibility, it algebraic-sign-izes using the conditional appearance probability which adopted the dependency (it considers as the "context" hereafter) of an input-statement character and the last alphabetic character.
[0021] The context is expressed with the tree structure as shown in drawing 5 . whenever the character string which passes along the alphabetic character of each node comes out -- the count of an appearance -- each node -- counting -- carrying out -- conditional -- a probability is searched for. In drawing 5 , the figure described on the right-hand of each alphabetic character shows the count of an appearance. For example, describing it as 5 on the right-hand of the alphabetic character (the die length of a branch is 1) a immediately under root Describing it as 2 on the right-hand of the alphabetic character a which means that the count of an appearance of the alphabetic character a is 5, and is under two steps from root (the die length of a branch is 2) Describing it as 1 on the right-hand of the alphabetic character a which means that the count of an appearance of an alphabetic character aa is 2, and is under three steps from root (the die length of a branch is 3) means that the count of an appearance of an alphabetic character aaa is 1.
[0022] Here, although the occurrence probability of all notations is defined according to the "context" which is the symbol string which appeared just before that notation, the die length of the symbol string used for formation of this context is called a "degree." There are following (1) and (2) in the context collection approach which is the method of a setup of a degree.
(1) How to make the degree of immobilization the conditions of the probability with a context of a fixed degree. for example, the context of the alphabetic character connected with two characters just before in the secondary context -- collection (drawing 5 the die length 3 of the branch from root) -- carrying out -- conditional -- Probability p (y|x1, x2) is acquired.
[0023] However, as for an attention coded character, x1, and x2, y means the 1st last character and the 2nd character, respectively, and p (y|x1, x2) means the probability for y to appear, after x1 and x2 continue and appear.
(2) The Blending context Blending (mixing of a degree) develops a degree according to input data, without fixing conditional-statement character queue length.
[0024] In the formation of a multiple-value algebraic sign, when there are many alphabetic characters which can appear, many alphabetic characters which do not appear at all exist in the compressed file corresponding (for example, when one character is expressed by 16 bits and the numbers of the alphabetic characters which can appear are 64K). In this case, when re-calculating the frequency of occurrence at every alphabetic character input, and performing algebraic-sign-ization dynamically, and the appearance possibility of a **** alphabetic character is considered and "1" is given to the initial value of each frequency of occurrence, many useless sections will be taken and compressibility will fall. As an approach of losing this futility, there is the primary [-] zero-order Blending approach. - The 1st order expresses what made the non-appeared alphabetic character same probability, and expresses the alphabetic character frequency of occurrence [zero-order] without the context.
[0025] - Explain the flow of the algebraic-sign-ized method using the primary zero-order Blending approach with reference to drawing 6 . In addition, this flow is processed with the equipment shown in drawing 3 . First, let upper limit =1, lower limit =0, and section width-of-face =1.0 be initial value in algebraic-sign-izing at step 601. Moreover, it registers with a dictionary by making into a non-appeared alphabetic character the alphabetic

character (information source) in which all appearances are possible.
[0026] and a single-character (referred to as k) input is carried out from an input string (step 602) -- every -- it distinguishes in a dictionary whether it appeared in the flow (step 603).
[0027] At step 603, when it had not appeared and is distinguished, the section for non-appeared alphabetic characters is algebraic-sign-ized (step 607), and the alphabetic character k is algebraic-sign-ized by making all sheep appearance alphabetic characters into same probability (step 608). Then, the frequency of occurrence of the input-statement character k is set to "1" with a counter, and the alphabetic character k is removed from a non-appeared alphabetic character (step 609).
[0028] On the other hand, when having appeared is distinguished at step 603, the alphabetic character k is algebraic-sign-ized (step 604). then, a counter -- the frequency of occurrence of an input-statement character -- "1" -- it increases (step 605). And a dictionary is rearranged in order of the frequency of occurrence (step 606).
[0029] The accumulation frequency of occurrence is updated after activation of step 606 and step 609 (step 610). Then, activation is repeated from step 602.
[0030]
[Problem(s) to be Solved by the Invention] According to each statistical alphabetic character frequency of occurrence, as mentioned above in the compression method (probability statistics mold compression method) which assigns short code length to the high alphabetic character of an appearance probability, there are what uses each symbol frequency of occurrence (probability model) as a fixed target, and a thing changed dynamically.
[0031] In case it restores, the former needs the probability model which operated and obtained the probability model or all the character strings which were set up beforehand, and needs to hold the previous frequency of occurrence with compressed data.
[0032] On the other hand, the latter can build the probability model which is the ecad coding method which inputs a character string, and which is alike, therefore re-calculates and uses a probability model, and did not need to hold a probability model beforehand, and was based on each data for compression. However, when the character string which compresses is short, enough dictionaries cannot be built and good compressibility is not obtained.
[0033] This invention was not made in view of such a situation, and when the file size for compression is not sufficient magnitude for probability-model construction, it makes it a technical problem to offer the data compression approach and equipment which can obtain good compressibility, without holding the discrete character frequency of occurrence beforehand.
[0034]
[Means for Solving the Problem]
The 1st data compression approach of <data compression approach of ** 1st of this invention> this invention is constituted as following, in order to solve the technical problem mentioned above (it corresponds to claim 1).
[0035] That is, in the data compression approach of performing variable length coding which outputs the code length according to an appearance probability, it has the group configuration step, the group appearance probability count step, the alphabetic character appearance probability count step in a group, and the input-statement character coding step.
[0036] A group configuration step is classified into two or more hierarchical groups, respectively for every alphabetic character which has the same statistical property for the alphabetic character which may be inputted mutually. A group appearance probability count step calculates the appearance probability of each of said group.
[0037] The alphabetic character appearance probability count step in a group calculates the appearance probability of the input-statement character in said two or more groups. An input-statement character coding step encodes an input-statement character based on the appearance probability calculated at said alphabetic character appearance probability count step in a group.
[0038] The 2nd data compression approach of <data compression approach of ** 2nd of this invention> this invention is constituted as following, in order to solve the technical problem mentioned above (it corresponds to claim 2).
[0039] That is, in the 1st data compression approach, at said group configuration step (S1), it fixes beforehand and a classification of the component of said group is given.
The 3rd data compression approach of <data compression approach of ** 3rd of this invention> this invention is constituted as following, in order to solve the technical problem mentioned above (it corresponds to claim 3).

[0040] That is, in the 1st data compression approach, at said group appearance probability count step (S2), it fixes beforehand and the appearance probability of said group is given.
The 4th data compression approach of <data compression approach of ** 4th of this invention> this invention is constituted as following, in order to solve the technical problem mentioned above (it corresponds to claim 4).
[0041] That is, in the 1st data compression approach, at said group appearance probability count step (S2), while setting initial value as the appearance probability of said group beforehand, the appearance probability of this group is dynamically re-calculated according to the input of said alphabetic character.
[0042] The 5th data compression approach of <data compression approach of ** 5th of this invention> this invention is constituted as following, in order to solve the technical problem mentioned above (it corresponds to claim 5).
[0043] That is, in the 1st data compression approach, the appearance probability of said group is calculated at said group appearance probability count step (S2) by the conditional group appearance probability on condition of each group to which two or more last characters belong appearing.
[0044] The 6th data compression approach of <data compression approach of ** 6th of this invention> this invention is constituted as following, in order to solve the technical problem mentioned above (it corresponds to claim 6).
[0045] That is, in the 1st data compression approach, said group configuration step (S1) constitutes said two or more hierarchical groups from the 1st group which consists of Takaide present probability alphabetic characters, and the 2nd group which consists of low appearance probability alphabetic characters.
[0046] The data compression equipment of <data compression equipment of this invention> this invention is constituted as following, in order to solve the technical problem mentioned above (it corresponds to claim 9). .
[0047] That is, in the data compression equipment which performs variable length coding which outputs the code length according to an appearance probability, it has the group configuration section, the group appearance probability count section, the alphabetic character appearance probability count section in a group, and the input-statement character coding section.
[0048] The group configuration section classifies into two or more hierarchical groups the alphabetic character which may be inputted for every alphabetic character which has the same statistical property mutually, respectively. The group appearance probability count section calculates the appearance probability of each of said group.
[0049] The alphabetic character appearance probability count section in a group calculates the appearance probability of the input-statement character in said two or more groups. The input-statement character coding section encodes an input-statement character based on the appearance probability calculated in said alphabetic character appearance probability count section in a group.
[0050]
[Function]
<an operation of the 1st data compression approach of this invention> -- the alphabetic character which may be inputted is first classified into two or more hierarchical groups according to a group configuration step for every alphabetic character which has the same statistical property mutually. And the appearance probability of each group is calculated at a group appearance probability count step. And at the alphabetic character appearance probability count step in a group, the appearance probability of the input-statement character in two or more groups is calculated. And at an input-statement character coding step, an input-statement character is encoded based on the appearance probability calculated at the alphabetic character appearance probability count step in a group.
[0051] <an operation of the 2nd data compression approach of this invention> -- in an operation of the 1st data compression approach, at a group configuration step, a classification of the component of a group is fixed beforehand and given.
[0052] <an operation of the 3rd data compression approach of this invention> -- in an operation of the 1st data compression approach, at a group appearance probability count step, the appearance probability of a group is fixed beforehand and given.
[0053] <an operation of the 4th data compression approach of this invention> -- in an operation of the 1st data compression approach, at a group appearance probability count step, while initial value is beforehand set as the appearance probability of a group, the appearance probability of this group is dynamically re-calculated

according to the input of an alphabetic character.
[0054] <an operation of the 5th data compression approach of this invention> -- in an operation of the 1st data compression approach, it is calculated at a group appearance probability count step by the conditional group appearance probability on condition of each group to which two or more last characters belong [the appearance probability of a group] appearing.
[0055] <an operation of the 6th data compression approach of this invention> -- in an operation of the 1st data compression approach, two or more hierarchical groups are constituted from the 1st group constituted in a Takaide present probability alphabetic character, and the 2nd group which consists of low appearance probability alphabetic characters by the group configuration step.
[0056] <an operation of the data compression equipment of this invention> -- the alphabetic character which may be inputted is first classified into two or more hierarchical groups according to the group configuration section for every alphabetic character which has the same statistical property mutually. And the appearance probability of each group is calculated in the group appearance probability count section. And in the alphabetic character appearance probability count section in a group, the appearance probability of the input-statement character in two or more groups is calculated. And based on the appearance probability calculated in the alphabetic character appearance probability count section in a group, an input-statement character is encoded in the input-statement character coding section.
[0057]
[Example] Hereafter, the example of this invention is explained with reference to a drawing.
<Configuration of example> drawing 7 shows the configuration of the algebraic-sign equipment of this example. It has the element of following (b) - (b), and algebraic-sign equipment is constituted, as shown in this drawing.
(b) The alphabetic character group classification section 10 which classifies into either the alphabetic character group 1, the alphabetic character group 2 and the alphabetic character group 3 the alphabetic character which inputs a character string and is contained in this character string. Here, the alphabetic character group 1 uses a hiragana as a component, the alphabetic character group 2 uses a tooth space, punctuation, and a line feed mark as a component, and the alphabetic character group 3 uses other alphabetic characters, for example, the kanji, as a component.
(b) The probability-model creation section 20 which inputs a character string while inputting the group number (1, 2, or 3) of the alphabetic character group which the alphabetic character group classification section 10 outputs, and outputs the ranking of the input-statement character in the alphabetic character frequency of occurrence and each group.
(c) Sign part 30 which asks for the accumulation frequency of occurrence of the coded character in the group of a coded character continuously, and encodes the section while asking for the accumulation frequency of occurrence of the group from the group number of a coded character and encoding the section. This sign part 30 inputs "ranking of the alphabetic character frequency of occurrence and the input-statement character in each group" from the probability-model creation section 20, and outputs an algebraic sign while it inputs "a group number and the group frequency of occurrence" from the alphabetic character group classification section 10.
[0058] Hereafter, said (**) - (Ha) an element are explained to a detail.
The [alphabetic character group classification section 10] The alphabetic character group classification section 10 consists of the group classification section 11 and a group probability attaching part 12, as shown in drawing 8 .
[0059] The group classification section 11 inputs a character string, classifies into either the alphabetic character group 1, the alphabetic character group 2 and the alphabetic character group 3 the alphabetic character (it is also called a symbol) contained in this character string, and outputs the group number of the classified alphabetic character group. The group classification section 11 has conversion table 11a which a symbol and a group number are made to correspond and stores them. The group number stored in this conversion table 11a is outputted to the probability-model creation section 20 and sign part 30.
[0060] The group probability attaching part 12 inputs a group number from the group classification section 11, and outputs the frequency of occurrence for every alphabetic character group. The group probability attaching part 12 has conversion table 12a which a group number and the probability for every group are made to correspond, and stores them. The group appearance probability stored in this conversion table 12a is outputted to sign part 30.
[0061] The [probability-model creation section 20] The probability-model creation section 20 consists of a

dictionary 21 and a counter 22. A dictionary 21 inputs the group number to which the alphabetic character inputted from the alphabetic character group classification section 10 belongs, and outputs group number ranking (frequency-of-occurrence ranking in a group) while it inputs a character string. And the dictionary 21 has conversion table 21a which a symbol and group number ranking are made to correspond and stores them for every alphabetic character group. The group number ranking stored in this conversion table 21a is outputted to sign part 30.

[0062] A counter 22 inputs group number ranking from a dictionary 21, and outputs an alphabetic character appearance probability. And the counter 22 has conversion table 22a which the frequency-of-occurrence ranking and the alphabetic character frequency of occurrence in a group are made to correspond, and stores them for every alphabetic character group.

[0063] [Sign part 30] Sign part 30 consists of a table 31 and the algebraic-sign-ized section 32. A table 31 inputs "the alphabetic character ranking in a group, and the alphabetic character appearance probability in a group" from the probability-model creation section 20 while inputting "a group number and a group appearance probability" from the alphabetic character group classification section 10. And the table 31 has table 31a which a group number and the accumulation frequency of occurrence are made to correspond, and stores them, and two or more table 31b which the alphabetic character ranking in a county and the accumulation frequency of occurrence are made to correspond, and stores them for every alphabetic character group.

[0064] The algebraic-sign-ized section 32 inputs the accumulation frequency of occurrence which a table 31 holds, and outputs an algebraic sign. Here, the information which alphabetic character belongs to which group, and the information about the frequency of occurrence of an alphabetic character group are given in first stage according to the frequency of occurrence expected beforehand. For example, as shown in drawing 9 , alphabetic characters, such as a tooth space (null), and E, T, are classified into a Takaide present alphabetic character group, and alphabetic characters, such as H, D, and L, are classified into a low appearance alphabetic character group. And each group appearance probability should take total of each appearance probability of the alphabetic character belonging to each group.

[0065] Actuation of an example is explained with reference to <actuation of an example>, next drawing 10 . First, it is upper limit =1, lower limit =0, and section width-of-face =1.0 as initial setting of the symbolic language algebraic-sign-ized at step 1001. It carries out.

[0066] Here, the alphabetic character group classification section 10 initializes the group classification of the group classification section 11, and the group probability of the group probability attaching part 12 based on the frequency of occurrence expected beforehand. In addition, initialization of a group classification is giving the information the component and which alphabetic character of each group belonging to which group, and initialization of a group probability is giving the appearance probability of the group 1:group 2:group 3= 3:5:1 and a group according to initial value.

[0067] And the probability-model creation section 20 is classified into each symbol group, prepares the counter 22 for every symbol, and initializes it to 1. Moreover, the probability-model creation section 20 calculates the ranking of each separate symbol, and the accumulation frequency of occurrence for every alphabetic character group while accumulating and calculating the group accumulation frequency of occurrence. In addition, it says adding the frequency of occurrence of a group 3 - Group M as accumulating and calculating the group accumulation frequency of occurrence, and considering as the accumulation frequency of occurrence of a group 2.

[0068] next, a single-character (referred to as "k") input is carried out from an input string (step 1002) -- every -- the alphabetic character group classification section 10 distinguishes the group (referred to as "K") to which the dictionary of the group classification section 11 is searched and an input-statement character belongs (step 1003).

[0069] Here, the probability-model creation section 20 searches a dictionary 21 based on the group and input-statement character which were distinguished at step 1003, and outputs frequency-of-occurrence ranking and each alphabetic character frequency of occurrence of a group.

[0070] and the algebraic-sign section 30 -- the alphabetic character group accumulation frequency of occurrence -- using it -- the alphabetic character group K -- an algebraic sign ---izing (step 1004) -- the input-statement character k is algebraic-sign-ized (step 1005). in addition, algebraic-sign-izing of step 1004 -- (**) -- asking for the upper limit and lower limit of the section of an input alphabetic character group using a group number and the

group accumulation frequency of occurrence, and (**) -- it is carried out more to asking for the upper limit and lower limit of the section of an input-statement character using the accumulation frequency of occurrence of the frequency-of-occurrence ranking in a group of an input-statement character, and this group, and outputting the any value of the section (Ha) as a sign.
[0071] and the counter 22 -- the frequency of occurrence of the input-statement character k -- "1" -- it increases (step 1006) and the dictionary of the alphabetic character group K is rearranged in order of frequency (step 1007). "1" -- in connection with the alphabetic character which increased, frequency-of-occurrence ranking and the accumulation frequency of occurrence are updated (step 1008). [next,] Then, activation is repeated from step 1002.
[0072] [Actuation of algebraic-sign-izing using the primary [-] zero-order Blending approach] Next, actuation of algebraic-sign-izing using the primary [-] zero-order Blending approach is explained with reference to drawing 11 .
[0073] first -- step 1101 -- as initial setting of algebraic-sign-izing -- (**) -- preparing the alphabetic character group accumulation frequency of occurrence and (**) -- it performs setting each alphabetic character frequency of occurrence to 0, registering a whole sentence character for every alphabetic character group as an alphabetic character non-appeared (Ha), and setting as 1 the non-appeared alphabetic character probability prepared for (d) each alphabetic character group of every.
[0074] next, a single-character (referred to as "k") input is carried out from an input string (step 1102) -- every -- the alphabetic character group classification section 10 distinguishes the group (referred to as "K") to which the dictionary of the group classification section 11 is searched and an input-statement character belongs (step 1103).

[0075] And the algebraic-sign-ized section 30 uses the alphabetic character group accumulation frequency of occurrence, and algebraic-sign-izes the alphabetic character group K (step 1104). Here, it is judged whether the alphabetic character group K had appeared previously (step 1105). the case where it is judged at step 1105 that it had appeared previously -- the accumulation frequency of occurrence of the alphabetic character group K -- using it -- the alphabetic character k -- an algebraic sign ---izing (step 1106) -- the alphabetic character k -- counting (step 1107) -- a dictionary is rearranged in order of frequency (step 1108).
[0076] the case where it is judged at step 1105 on the other hand that it has not appeared previously -- the non-appeared alphabetic character section of the alphabetic character group K -- an algebraic sign ---izing (step 1109) -- the alphabetic character k -- an algebraic sign ---izing (step 1110) -- the alphabetic character k is inserted in the dictionary of the alphabetic character group K, and the alphabetic character k is removed from the non-appeared alphabetic character of the alphabetic character group K (step 1111). In addition, at step 1110, all the sheep appearance alphabetic characters of the alphabetic character group K are made into same probability.
[0077] After step 1108 and step 1111, the accumulation frequency of occurrence of the alphabetic character group K is updated.
[Example of algebraic-sign-izing of sign part 30] Drawing 12 is drawing showing the example of algebraic-sign-izing of sign part 30. In drawing 12 , a "hiragana" is made into the alphabetic character group 1, and "a tooth space, punctuation, and a line feed mark" are made into the alphabetic character group 3 for the "figure" etc. of the alphabetic character group 2 and others. It is considering as the single-character group. The appearance probability of a "hiragana" is 0.52 and the appearance probability of "a tooth space, punctuation, and a line feed mark" is 0.13. In early stages of compressive, no alphabetic character has appeared and the frequency of occurrence of each alphabetic character is 0.
[0078] in this case -- the appearance by same probability is possible for every alphabetic character in the conventional method -- thinking -- etc. -- although the sign section of width of face is set up, in this example, as shown in drawing 12 (B), it sets according to the appearance probability of each group, and considers as width of face [alphabetic character / which belongs to that alphabetic character group in the alphabetic character group section / each]. Each alphabetic character group section is divided according to each group appearance probability (refer to drawing 12 (A)) mentioned above.
[0079] According to the method which appoints each alphabetic character section after appointing the alphabetic character group section of this invention, as shown in drawing 12 (B), the large sign section can be given from the phase in early stages of compression to the high alphabetic character of an appearance probability.
[0080] In the <modification of this example> aforementioned example, although it came considering the

alphabetic character group appearance probability as a fixed thing, the modification which changes an alphabetic character group appearance probability dynamically is stated.
[0081] (1) the thing which changes dynamically the conditional appearance probability which took in the context of the thing (2) group which changes the appearance probability of each group for an alphabetic character group appearance probability dynamically and which changes the appearance probability of each group for an alphabetic character group appearance probability dynamically is first shown in drawing 13 . This is equivalent to the alphabetic character group classification section 10 in drawing 7 . the alphabetic character group classification section 10 gives initial value to each alphabetic character group, and inputs an alphabetic character as the group classification section 11 which shows which alphabetic character belongs to which group -- ** -- the frequency of occurrence of the group to which is resembled and the alphabetic character belongs -- "1" -- it increases and consists of group counters 13 which update the group accumulation frequency of occurrence.
[0082] The actuation is explained with reference to drawing 14 . First, at step 1401, as initial setting, the alphabetic character group accumulation frequency of occurrence is taken, each alphabetic character frequency of occurrence is set to 1, and the accumulation frequency of occurrence is taken for every alphabetic character group.
[0083] next, a single-character ("k") input is carried out from an input string (step 1402) -- every -- the alphabetic character group classification section 10 distinguishes the group (referred to as "K") to which the dictionary of the group classification section 11 is searched and an input-statement character belongs (step 1403).
[0084] and the algebraic-sign section 30 -- the alphabetic character group accumulation frequency of occurrence -- using it -- the alphabetic character group K -- an algebraic sign ---izing (step 1404) -- the input-statement character k is algebraic-sign-ized (step 1405).
[0085] And the frequency of occurrence of the alphabetic character k and the frequency of occurrence of the alphabetic character group K are made to increase by every [1], respectively (step 1406), and the dictionary of the alphabetic character group K is rearranged in order of frequency (step 1407).
[0086] Next, the accumulation frequency of occurrence of the alphabetic character group K is updated with the increment in step 1406 (step 1408). Similarly, the context of a group can be taken in and a conditional appearance probability can also be acquired dynamically. A zero-order value gives initial value, and as shown in drawing 5 , whenever the alphabetic character group which passes along each node group comes more than out of the primary relative probability, relative probability is called for by carrying out counting of the count of an appearance in each node. Conventionally, the group is each wooden joint by this example to the symbol having become each wooden joint here.
[0087] The flow in the case of taking the primary conditional appearance probability to the group frequency of occurrence is shown in drawing 15 . First, following (**) - (**) are performed as initialization (step 1601).
(b) Initialize each alphabetic character group frequency of occurrence.
(b) Take the alphabetic character group accumulation frequency of occurrence.
(c) Set each alphabetic character frequency of occurrence to "1."
(d) Take the accumulation frequency of occurrence for every alphabetic character group.
Even (e) holds the group number of a front alphabetic character.
The register R (= context) with which even (**) holds the front group number is initialized.
[0088] Next, a single character (referred to as k) is inputted (step 1602). And it distinguishes to which alphabetic character group (referred to as K) the input-statement character k belongs (step 1603).
[0089] and conditional [which means "the frequency of occurrence of the frequency of occurrence/R of RK"] -- Probability P (K|R) is algebraic-sign-ized by sign part 30. That is, R The section is divided according to the probability for each group to happen continuously, among these it is Group K. The section is chosen. In addition, the minimum can be found with the accumulation frequency of occurrence of the alphabetic character group to which the section of each group happens following R (step 1604).
[0090] and conditional -- while algebraic-sign-izing Probability P (k|K), the conditional accumulation frequency of occurrence CF of an alphabetic character group (k|K) is used, and the input-statement character k is algebraic-sign-ized (step 1605).
[0091] And only "1" makes the value of the alphabetic character frequencies of occurrence C (k|K) and C (K|R) increase, respectively (step 1606). And the dictionary of the alphabetic character group K is rearranged according to the alphabetic character frequency of occurrence C (x|K) (step 1607).

[0092] And while updating the alphabetic character accumulation frequency of occurrence CF of the alphabetic character group K (x|K), the group accumulation frequency of occurrence CF of the alphabetic character group following the alphabetic character group R (X|R) is updated (step 1608). And the input-statement character k is set as Register R (step 1609).
[0093] Henceforth, the processing from step 1602 is repeated.
<The effectiveness of an example>, next the data compression effectiveness of an example are explained with reference to drawing 16 .
[0094] Drawing 16 (A) shows three cases in the case of depending it on an ecad coding method, when are based on this example, and depending on a static coding method [how a data compression rate changes with the sizes of the file for compression] (semi- ecad). The axes of abscissa and axes of abscissa of drawing 16 (A) are a file size for compression, and a data compression rate, respectively, and when line 7a is based on this example and line 7b is based on a static coding method, line 7c shows the case where it is based on an ecad coding method, respectively.
[0095] When being based on a static coding method so that clearly from drawing 16 (A), it turns out that the almost fixed data compression rate was held and the data compression has been carried out most in for a comparison irrespective of the size of the file for compression. On the other hand, when based on an ecad coding method and this example, it turns out that a data compression is small improved compressibility more, so that the size of the file for compression becomes large, and the data compression rate of a static coding method is approached. And when based on this example, the data compression rate is always small rather than the case where it is based on an ecad coding method.
[0096] The difference of the data compression rate in the case of being based on the case where it is based on an ecad coding method in case the file size for compression is about 0 here, and a static coding method is because the initial value of each alphabetic character frequency of occurrence is given to the static coding method.
[0097] Moreover, the difference of a data compression rate with the case where it is based on the case where it is based on an ecad coding method in case the file size for compression is **** 0, and this example is because the initial value of each group frequency of occurrence is given to this example.
[0098] Next, drawing 16 (B) compares about three cases in the case of being based on an ecad coding method, when the file size before compression twists to this example how the file size after compression changes, and based on a static coding method (semi- ecad). In addition, also when not encoding in drawing 16 (B), it has described at reference. The axes of abscissa and axes of ordinate of drawing 16 (B) are a file size before compression, and a file size after compression, respectively, and when 7d of lines is based on this example, line 7e is based on a static coding method, and 7f of lines is based on an ecad coding method, 7g of lines shows the case where it does not encode, respectively.
[0099] It turns out that the increment in the file size after compression becomes blunt, so that from drawing 16 (B) and the file size before compression becomes large in any [which encodes] case. Moreover, when are based on the case where it is based on a static coding method, and this example and a file size is small, it turns out that it becomes larger than the file size before the file size after compression compressing.
[0100] And when the file size before compression is smaller than a predetermined value, and it is based on a static coding method, and based on this example, the file size after compressing in order in the case of being based on an ecad coding method becomes small, but if the file file size before compression becomes large rather than said predetermined value, when are based on an ecad coding method and based on a static coding method, it understands that the file size after compressing in order in the case of being based on this example becomes small.
[0101] It is because it has the initial value information on each alphabetic character frequency of occurrence as an auxiliary data that the file size at the time of adding an auxiliary data to a static coding method in case the file size before compression is about 0 here is not 0.
[0102] Moreover, it is because it has the initial value information on each group frequency of occurrence as an auxiliary data that the file size at the time of adding an auxiliary data to this example in case the file size before compression is about 0 is not 0.
[0103]
[Effect of the Invention] According to the 1st data compression approach of this invention, and data compression equipment, an alphabetic character is classified into two or more groups for every alphabetic character which has

the same statistical property mutually, and further, since the appearance probability of each group was calculated, compared with the conventional approach, the optimal sign field can be assigned in an early phase. There are many alphabetic characters which can appear, and this has it, especially when the file for compression is small. [effective] That is, by the conventional ecad coding method, the input train of a certain amount of die length is needed for building a probability model, and when the size for compression is small, by this invention, sufficient compressibility can be obtained to compressibility not increasing.

[0104] Since the sign according to the appearance probability of data can be beforehand assigned compared with the 1st data compression approach according to the 2nd of this invention, and the 3rd data compression approach, also when a file size is small, it becomes possible to obtain high compressibility.

[0105] According to the 4th data compression approach of this invention, since the frequency of occurrence is recalculated according to input data, the compression based on the frequency of occurrence gradually based on data is attained. According to the 5th of this invention, and the 6th data compression approach, still higher compressibility comes to be obtained by using the relative probability on condition of the group to which the alphabetic character which appeared before the group to which the alphabetic character which appeared immediately before belongs, or personally belongs.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the principle Fig. of the data compression of this invention. (A) shows the principle Fig. of the data compression approach, and (B) shows the principle Fig. of data compression equipment.
[Drawing 2] It is drawing showing the principle of a multiple-value algebraic sign. (A) shows the frequency of occurrence of each alphabetic character. (B) shows the accumulation frequency of occurrence of the order of the frequency of occurrence. (C) shows the principle of algebraic-sign-izing.
[Drawing 3] It is drawing showing the equipment configuration of algebraic-sign-izing.
[Drawing 4] It is drawing showing the flow of the conventional formation of a multiple-value algebraic sign.
[Drawing 5] It is drawing showing the tree structure (in the secondary case) of the context.
[Drawing 6] It is drawing showing the flow of the conventional formation of a multiple-value algebraic sign (-1, zero-order blending).
[Drawing 7] It is drawing showing the outline of the equipment configuration of an example.
[Drawing 8] It is drawing showing the equipment configuration of an example in a detail.
[Drawing 9] It is drawing showing a group classification and a group appearance probability.
[Drawing 10] It is drawing showing the flow of the formation of a multiple-value algebraic sign of an example (the 1).
[Drawing 11] It is drawing showing the flow of the formation of a multiple-value algebraic sign of an example (the 2). This flow is -1 and zero-order blending.
[Drawing 12] It is drawing showing an alphabetic character group appearance probability and the initial sign section. (A) shows the alphabetic character group appearance probability. (B) shows the sign section at the probability-model non-held section time.
[Drawing 13] It is drawing showing the alphabetic character group classification section.
[Drawing 14] It is drawing showing the flow of the formation of a multiple-value algebraic sign of an example (the 3).
[Drawing 15] It is drawing showing the flow of the formation of a multiple-value algebraic sign of an example (the 4).
[Drawing 16] It is the comparison Fig. of the effectiveness of the conventional algebraic-sign-izing and algebraic-sign-izing of this example. (A) shows change of the data compression rate at the time of changing the file size for compression. (B) shows change of the file size after the compression at the time of changing the file size before compression.
[Description of Notations]
S1 Group configuration step
S2 Group appearance probability count step
S3 Alphabetic character appearance probability count step in a group
M1 Group configuration section
M2 Group appearance probability count section
M3 The alphabetic character appearance probability count section in a group
10 Alphabetic Character Group Classification Section
11 Group Classification Section
12 Group Probability Attaching Part
13 Group Counter

20 Probability-Model Creation Section
21 Dictionary
22 Counter
30 Sign Part
31 Table
32 Algebraic-Sign-ized Section
40 Algebraic-Sign Section

[Translation done.]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平8－167852

(43)公開日　平成8年(1996)6月25日

(51)Int.Cl.[6]	識別記号	庁内整理番号	FI	技術表示箇所
H03M 7/40		9382－5K		
G06F 5/00		H		

審査請求　未請求　請求項の数7　OL　（全 18 頁）

(21)出願番号　特願平6－308662

(22)出願日　平成6年(1994)12月13日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　佐藤　宜子
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　岡田　佳之
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　吉田　茂
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74)代理人　弁理士　遠山　勉　（外１名）

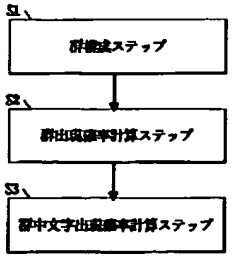
(54)【発明の名称】　データ圧縮方法及び装置

(57)【要約】

【目的】　圧縮対象ファイルサイズが確率モデル構築に十分な大きさでない場合に、予め個々の文字出現頻度を保持せずに、良い圧縮率を得ることができるデータ圧縮方法及び装置を提供することを目的とする。

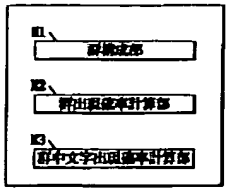
【構成】　入力される可能性がある文字を、互いに同じ統計的性質を有する文字毎に階層的な複数の群にそれぞれ分類する群構成ステップと、それぞれの群の出現確率を計算する群出現確率計算ステップと、複数の群中における入力文字の出現確率を計算する群中文字出現確率計算ステップと、群中文字出現確率計算ステップで計算された出現確率に基づいて入力文字を符号化する入力文字符号化ステップとを備えて構成した。

本発明によるデータ圧縮方法の原理図



(A)

本発明によるデータ圧縮装置の原理図



(B)

1

【特許請求の範囲】

【請求項１】入力された文字を、その出現確率に応じた符号長を持つ可変長符号に符号化することでデータの圧縮を行うデータ圧縮方法において、

入力される可能性がある文字を、互いに同じ統計的性質を有する文字毎に階層的な複数の群にそれぞれ分類する群構成ステップと、

前記それぞれの群の出現確率を計算する群出現確率計算ステップと、

前記複数の群中における入力文字の出現確率を計算する群中文字出現確率計算ステップと、

前記群中文字出現確率計算ステップで計算された出現確率に基づいて入力文字を符号化する入力文字符号化ステップとを備えたことを特徴とするデータ圧縮方法。

【請求項２】前記群構成ステップでは、前記群の構成要素の分類を予め固定して与えることを特徴とする請求項１に記載のデータ圧縮方法。

【請求項３】前記群出現確率計算ステップでは、前記群の出現確率を予め固定して与えることを特徴とする請求項１に記載のデータ圧縮方法。

【請求項４】前記群出現確率計算ステップでは、前記群の出現確率に予め初期値を設定するとともに、この群の出現確率を前記文字の入力に応じて動的に再計算することを特徴とする請求項１に記載のデータ圧縮方法。

【請求項５】前記群出現確率計算ステップでは、前記群の出現確率を、直前の複数文字が属する各々の群が出現することを条件とする条件付群出現確率で計算することを特徴とする請求項１に記載のデータ圧縮方法。

【請求項６】前記群構成ステップでは、前記階層的な複数の群を、高出現確率文字で構成される第１の群と、低出現確率文字で構成される第２の群とで構成することを特徴とする請求項１に記載のデータ圧縮方法。

【請求項７】入力された文字を、その出現確率に応じた符号長を持つ可変長符号に符号化することでデータの圧縮を行うデータ圧縮装置において、

入力される可能性がある文字を、互いに同じ統計的性質を有する文字毎に階層的な複数の群にそれぞれ分類する群構成部と、

前記それぞれの群の出現確率を計算する群出現確率計算部と、

前記複数の群中における入力文字の出現確率を計算する群中文字出現確率計算部と、

前記群中文字出現確率計算部で計算された出現確率に基づいて入力文字を符号化する入力文字符号化部とを備えたことを特徴とするデータ圧縮装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】近年、文字コード、画像データ等の様々なデータがコンピュータで扱われるようになるのに伴い、取り扱われるデータ量も増大している。そのよ

2

うな大量のデータは、データ中の冗長な部分を省いて圧縮することにより、記憶容量を減らしたり、早く伝送したりできるようになる。

【０００２】一方、圧縮を行ったデータは、参照・利用する際に復元する必要があるため、圧縮する前のデータに比べアクセス速度が低下する。そこで、これまでのデータ圧縮は、主に、一部参照を行うことが希な、データのバックアップや通信のときにのみ利用されている。

【０００３】しかし、近年では、圧縮専用ＬＳＩが利用されるようになったため、圧縮データの復元速度は短くなり、通常のデータと同様にアクセスを行うデータに対しても、圧縮・復元を行うことが考えられてきている。

【０００４】そこで、圧縮を行った場合、圧縮データ単位毎に復元を行うため、アクセスするデータサイズ単位と、圧縮するデータサイズは同程度（５Ｋｂｙｔｅ以下、１～２Ｋｂｙｔｅ程度）で行うことが望まれている。

【０００５】

【従来の技術】様々な種類のデータ（文字コード、画像データ等）に適用できるデータ圧縮方式として、ユニバーサル符号化方式が提案されている。ここで、本発明は、文字コードの圧縮に限定されず、様々なデータに適用できるが、以下では、情報理論に基づき、データの１ワード単位を文字（アルファベット）と呼び、データが任意ワードつながったものを文字列と呼ぶことにする。

【０００６】ユニバーサル符号化方式の中で代表的な方式として、算術符号化方式がある。この方式は、従来よく使われているハフマン符号のように、１文字づつばらばらに符号化の１点に対応付け、２進数の小数点以下を符号として出力するものである。

【０００７】ここで、多値算術符号化の原理を、図２を参照して説明する。まず、算術符号では、文字列を実数０と１の間（［０，１））のある実数の区間を用いて表すということが基本アイデアになっている。

【０００８】ここで、区間［０，１）を採用するのは、２進数の小数点以下を符号として出力するためである。また、以上“［”と未満“）”となっている理由は、［０，１］では、０と１の小数点以下が同じになって０と１を区別できなくなるためであり、（０、１）では、値としての０が使用できなくなるためである。

【０００９】図２（Ａ）は、ａ，ｂ，ｃ，ｄの４文字が出現すると仮定した場合に、各々の文字の出現頻度を示している。図２（Ａ）中、横軸の文字ａ，ｂ，ｃ，ｄの下側に記された（４）、（２）、（１）、（３）は、それぞれの文字の出現頻度順位を示している。

【００１０】図２（Ａ）に示された各文字の出現頻度に基づいて、文字毎の累積出現頻度確率を出現頻度順に示したのが図２（Ｂ）である。即ち、図２（Ｂ）中、横軸にｃｆ０と記された列は、ｃ，ｂ，ｄ，ａの４文字中の文字ｃの累積出現頻度確率を示している。同様に、横軸

にｃｆ１と記された列は、ｂ，ｄ，ａの３文字中の文字ｂの累積出現頻度確率を示している。同様に、横軸にｃｆ２と記された欄は、ｄ，ａの２文字中の文字ｄの累積出現頻度確率を示している。同様に、横軸にｃｆ３と記された欄は、文字ａの累積出現頻度確率を示している。

【００１１】そして、図２（Ｂ）に示された累積出現頻度確率から算術符号化を行う方法を示したのが図２（Ｃ）である。即ち、文字ｃを入力した段階で、対応する区間１０として、文字ｃの累積出現頻度確率（図２（Ｂ）においてｃｆ０と記された列で斜線が付された箇所）と同等の区間幅を採用する。

【００１２】次に、２番目の文字ａが入力された段階

＊で、対応する区間１１として、１文字目に対応する区間１０を各文字の累積出現頻度確率で再分割して得られる区間１１を採用する。

【００１３】そして、３番目の文字ｄが入力された段階で、対応する区間１２として、２文字目に対応する区間１１を各文字の累積出現頻度確率で再分割して得られる区間１２を採用する。

【００１４】このようにして、文字列ａｃｄは区間１２の任意値（区間１２の上端と下端の間の任意の値）として符号化される。ここで、各区間の下端は、式（１－１）～（１－２）により求められる。

新たな部分区間の下端＝現部分区間の下端＋現部分区間幅×注目文字の累積確率

・・・（１－１）

新たな部分区間幅　＝現部分区間幅×注目文字の確率

・・・（１－２）

なお、符号語を復元するには、符号語が各文字の確率に分けたどの区間に含まれるか、逐次再分割しながら調べればよい。

【００１５】このように、算術符号化では、区間で符号化するが、復号する過程では実際に区間が与えられる必要は無く、区間の中のある一つの数が指定されればよい。具体的な符号語としては、区間内の数の中でできる※

※だけ短いビット数で表せるものを選べばよい。

【００１６】即ち、出現頻度が高いほど区間幅が大きくなるということから、区間幅が大きいほど小数点以下の数が少なくなり、短いビット数で表せるようになる。以上は、各シンボル出現頻度を固定した例の説明であるが、以下に示すように、出現頻度（確率モデル）を逐次変更して、動的に行うこともできる。

注目文字の累積確率＝注目文字より出現頻度の低い文字の出現回数の累積／入力文字列長

・・・（２－１）

注目文字の確率＝注目文字の出現回数／入力文字列長

・・・（２－２）

ここで、出現頻度を文字入力の度に再計算する動的な算術符号化を行う装置の構成を図３に示す。この装置は、図２（Ａ）のような入力された文字の出現頻度を作成すると共に、図２（Ｂ）のような文字ごとの出現頻度順の累積出現頻度を作成する確率確率モデル（シンボル出現頻度）作成部２０と、図２（Ｃ）のような累積出現頻度確率から算術符号化を行う算術符号部４０とから構成されている。そして、確率モデル作成部２０は、図示していない辞書とカウンタとを有している。

【００１７】次に、図４のフローにより、図３の算術符号化装置の動作を説明する。まず、ステップ４０１では、上端＝１、下端＝０、区間幅＝１．０を算術符号化の初期値とする。このとき、確率モデル作成部２０の辞書は、シンボルと出現頻度順位を保持し、カウンタは各シンボル出現頻度を保持する。また、初期化として、シンボル数（出現が考えられる文字数：１ｂｙｔｅの時２５６）の辞書を準備し、各文字ごとに出現頻度をカウントするカウンタを準備し“１”に初期化する。そして、算術符号部４０は、各シンボルの順位、また、累積出現頻度を保持する。

【００１８】入力文字列より一文字（ｋとする）入力す★る（ステップ４０２）毎に、辞書より出現頻度順位を選択し、この番号及び累積出現頻度を用いて算術符号部４０にて区間を計算し、算術符号化する（ステップ４０３）。つまり、入力文字の区間の上端と下端を式（１－１）～（１－２）及び式（２－１）～（２－２）に基づいて求め、区間の任意の値を符号として出力する。

【００１９】その後、カウンタにて入力文字の出現頻度を“１”増やす（ステップ４０４）。“１”増加した文字に伴い、頻度順に辞書を並び替える（ステップ４０５）と共に、累積出現頻度を更新する（ステップ４０６）。なお、ステップ４０５とステップ４０６は、逆に処理してもよい。

【００２０】以上のステップ４０２からステップ４０６までの操作は、繰り返し実行される。これまでの説明は、一文字毎の出現確率に基づいて算術符号化する場合の例である。更に圧縮率を高めるには、入力文字と直前の文字との依存関係（以下、「文脈」とする）を取り入れた、条件付出現確率を用いて算術符号化する。

【００２１】文脈は、図５に示すように、木構造で表される。各ノードの文字を通る文字列が出る毎に出現回数を各ノードにて計数しておいて条件付き確率を求める。

図5において、各文字の右隣に記された数字が出現回数を示している。例えば、rootの直ぐ下にある（枝の長さが1）文字aの右隣に5と記されているのは、文字aの出現回数が5であることを意味し、rootから2段下にある（枝の長さが2）文字aの右隣に2と記されているのは、文字aaの出現回数が2であることを意味し、rootから3段下にある（枝の長さが3）文字aの右隣に1と記されているのは、文字aaaの出現回数が1であることを意味している。

【0022】ここで、全ての記号の生起確率は、その記号の直前に出現した記号列である「文脈」に従って定められるが、この文脈の形成に利用される記号列の長さは「次数」と呼ばれる。次数の設定の仕方である文脈収集方法には、下記の（1）及び（2）がある。

（1）固定次数の文脈

条件付確率の条件を固定の次数にする方法。例えば、2次の文脈では直前2文字につながる文字の文脈を収集（図5では、rootからの枝の長さ3）し、条件付き確率$p(y|x1,x2)$を得る。

【0023】ただし、yは注目符号化文字、x1、x2はそれぞれ直前の第1文字、第2文字を意味し、$p(y|x1,x2)$は、x1、x2が続いて出現した後に、yが出現する確率を意味している。

（2）Blending文脈

Blending（次数の混合）は、条件文字列の長さを固定せずに、入力データに応じて次数を伸ばす。

【0024】多値算術符号化において出現可能な文字数が多い場合（例えば、1文字が16bitで表現され、出現可能な文字が64K個の場合）には、該当する圧縮ファイルに全く出現しない文字が多数存在する。この場合、出現頻度を文字入力の度に再計算して動的に算術符号化を行うときに、全各文字の出現可能性を考えて各出現頻度の初期値に“1”を与えると、無駄な区間を多くとり、圧縮率が低下することになる。この無駄をなくす方法として、－1次と0次のBlending方法がある。－1次は、未出現文字を等確率にしたものをあらわし、0次は、文脈無しの文字出現頻度を表す。

【0025】－1次、0次のBlending方法を用いた算術符号化方式のフローを、図6を参照して説明する。なお、このフローは、例えば図3に示す装置で処理される。まず、ステップ601では、算術符号化にあたって、上端＝1、下端＝0、区間幅＝1．0を初期値とする。また、全出現可能な文字（情報源）を未出現文字として辞書に登録する。

【0026】そして、入力文字列より一文字（kとする）入力する（ステップ602）毎に、それがフロー中に出現したかどうかを辞書により判別する（ステップ603）。

【0027】ステップ603で、出現していないと判別された場合は、未出現文字用区間を算術符号化し（ステップ607）、全未出現文字を等確率として文字kを算術符号化する（ステップ608）。その後、カウンタにて入力文字kの出現頻度を“1”とし、文字kを未出現文字より除く（ステップ609）。

【0028】一方、ステップ603で、出現していたと判別された場合は、文字kを算術符号化する（ステップ604）。その後、カウンタにて入力文字の出現頻度を“1”増やす（ステップ605）。そして、出現頻度順に辞書を並び替える（ステップ606）。

【0029】ステップ606とステップ609の実行後に、累積出現頻度を更新する（ステップ610）。その後、ステップ602から実行を繰り返す。

【0030】

【発明が解決しようとする課題】統計的な各文字出現頻度に従い、出現確率の高い文字に対して短い符号長を割り振る圧縮方式（確率統計型圧縮方式）において、前述したように、各シンボル出現頻度（確率モデル）を固定的にするものと、動的に変更するものとがある。

【0031】前者は、復元する際に予め設定した確率モデルまたは全文字列を操作して得た確率モデルを必要とし、圧縮したデータとともに、先の出現頻度を保持する必要がある。

【0032】一方、後者は、文字列を入力するに従って、確率モデルを再計算して使う適応型符号化方式であり、予め確率モデルを保持しなくてよく、また、各圧縮対象データに即した確率モデルを構築することができる。しかし、圧縮を行う文字列が短い場合には、十分な辞書を構築することができず、良い圧縮率が得られない。

【0033】本発明は、このような事情に鑑みてなされたもので、圧縮対象ファイルサイズが確率モデル構築に十分な大きさでない場合に、予め個々の文字出現頻度を保持せずに、良い圧縮率を得ることができるデータ圧縮方法及び装置を提供することを課題とする。

【0034】

【課題を解決するための手段】

＜本発明の第1のデータ圧縮方法＞本発明の第1のデータ圧縮方法は、前述した課題を解決するため、下記の如く構成されている（請求項1に対応）。

【0035】即ち、出現確率に応じた符号長を出力する可変長符号化を行うデータ圧縮方法において、群構成ステップと、群出現確率計算ステップと、群中文字出現確率計算ステップと、入力文字符号化ステップとを備えている。

【0036】群構成ステップは、入力される可能性がある文字を、互いに同じ統計的性質を有する文字毎に階層的な複数の群にそれぞれ分類する。群出現確率計算ステップは、前記それぞれの群の出現確率を計算する。

【0037】群中文字出現確率計算ステップは、前記複数の群中における入力文字の出現確率を計算する。入力

文字符号化ステップは、前記群中文字出現確率計算ステップで計算された出現確率に基づいて入力文字を符号化する。

【００３８】<本発明の第２のデータ圧縮方法>本発明の第２のデータ圧縮方法は、前述した課題を解決するため、下記の如く構成されている（請求項２に対応）。

【００３９】即ち、第１のデータ圧縮方法において、前記群構成ステップ（Ｓ１）では、前記群の構成要素の分類を予め固定して与える。

<本発明の第３のデータ圧縮方法>本発明の第３のデータ圧縮方法は、前述した課題を解決するため、下記の如く構成されている（請求項３に対応）。

【００４０】即ち、第１のデータ圧縮方法において、前記群出現確率計算ステップ（Ｓ２）では、前記群の出現確率を予め固定して与える。

<本発明の第４のデータ圧縮方法>本発明の第４のデータ圧縮方法は、前述した課題を解決するため、下記の如く構成されている（請求項４に対応）。

【００４１】即ち、第１のデータ圧縮方法において、前記群出現確率計算ステップ（Ｓ２）では、前記群の出現確率に予め初期値を設定するとともに、この群の出現確率を前記文字の入力に応じて動的に再計算する。

【００４２】<本発明の第５のデータ圧縮方法>本発明の第５のデータ圧縮方法は、前述した課題を解決するため、下記の如く構成されている（請求項５に対応）。

【００４３】即ち、第１のデータ圧縮方法において、前記群出現確率計算ステップ（Ｓ２）では、前記群の出現確率を、直前の複数文字が属する各々の群が出現することを条件とする条件付群出現確率で計算する。

【００４４】<本発明の第６のデータ圧縮方法>本発明の第６のデータ圧縮方法は、前述した課題を解決するため、下記の如く構成されている（請求項６に対応）。

【００４５】即ち、第１のデータ圧縮方法において、前記群構成ステップ（Ｓ１）では、前記階層的な複数の群を、高出現確率文字で構成される第１の群と、低出現確率文字で構成される第２の群とで構成する。

【００４６】<本発明のデータ圧縮装置>本発明のデータ圧縮装置は、前述した課題を解決するため、下記の如く構成されている（請求項９に対応）。。

【００４７】即ち、出現確率に応じた符号長を出力する可変長符号化を行うデータ圧縮装置において、群構成部と、群出現確率計算部と、群中文字出現確率計算部と、入力文字符号化部とを備えている。

【００４８】群構成部は、入力される可能性がある文字を、互いに同じ統計的性質を有する文字毎に階層的な複数の群にそれぞれ分類する。群出現確率計算部は、前記それぞれの群の出現確率を計算する。

【００４９】群中文字出現確率計算部は、前記複数の群中における入力文字の出現確率を計算する。入力文字符号化部は、前記群中文字出現確率計算部で計算された出現確率に基づいて入力文字を符号化する。

【００５０】

【作用】

<本発明の第１のデータ圧縮方法の作用>まず、群構成ステップでは、入力される可能性がある文字が、互いに同じ統計的性質を有する文字毎に階層的な複数の群に分類される。そして、群出現確率計算ステップでは、それぞれの群の出現確率が計算される。そして、群中文字出現確率計算ステップでは、複数の群中における入力文字の出現確率が計算される。そして、入力文字符号化ステップでは、群中文字出現確率計算ステップで計算された出現確率に基づいて入力文字が符号化される。

【００５１】<本発明の第２のデータ圧縮方法の作用>第１のデータ圧縮方法の作用において、群構成ステップでは、群の構成要素の分類が予め固定して与えられる。

【００５２】<本発明の第３のデータ圧縮方法の作用>第１のデータ圧縮方法の作用において、群出現確率計算ステップでは、群の出現確率が予め固定して与えられる。

【００５３】<本発明の第４のデータ圧縮方法の作用>第１のデータ圧縮方法の作用において、群出現確率計算ステップでは、群の出現確率に予め初期値が設定されるとともに、この群の出現確率が文字の入力に応じて動的に再計算される。

【００５４】<本発明の第５のデータ圧縮方法の作用>第１のデータ圧縮方法の作用において、群出現確率計算ステップでは、群の出現確率が、直前の複数文字が属する各々の群が出現することを条件とする条件付群出現確率で計算される。

【００５５】<本発明の第６のデータ圧縮方法の作用>第１のデータ圧縮方法の作用において、群構成ステップでは、階層的な複数の群が、高出現確率文字で構成される第１の群と、低出現確率文字で構成される第２の群とで構成される。

【００５６】<本発明のデータ圧縮装置の作用>まず、群構成部では、入力される可能性がある文字が、互いに同じ統計的性質を有する文字毎に階層的な複数の群に分類される。そして、群出現確率計算部では、それぞれの群の出現確率が計算される。そして、群中文字出現確率計算部では、複数の群中における入力文字の出現確率が計算される。そして、入力文字符号化部で、群中文字出現確率計算部で計算された出現確率に基づいて入力文字が符号化される。

【００５７】

【実施例】以下、本発明の実施例を図面を参照して説明する。

<実施例の構成>図７は、本実施例の算術符号装置の構成を示す。算術符号装置は、同図に示されるように、以下の（イ）～（ロ）の要素を備えて構成される。

（イ）文字列を入力し、該文字列に含まれる文字を、文

字群１、文字群２及び文字群３のいずれかに分類する文字群分類部１０。ここで、文字群１は、ひらがなを構成要素とし、文字群２は、スペース、句読点及び改行マークを構成要素とし、文字群３は、その他の文字、例えば漢字を構成要素とする。

（ロ）文字群分類部１０が出力する文字群の群番号（１、２、３のいずれか）を入力すると共に文字列を入力し、文字出現頻度と各群における入力文字の順位を出力する確率モデル作成部２０。

（ハ）符号化文字の群番号からその群の累積出現頻度を求め、その区間を符号化すると共に、続いて符号化文字のその群における符号化文字の累積出現頻度を求め、その区間を符号化する符号部３０。この符号部３０は、文字群分類部１０から「群番号及び群出現頻度」を入力すると共に、確率モデル作成部２０から「文字出現頻度及び各群における入力文字の順位」を入力し、算術符号を出力する。

【００５８】以下、前記（イ）～（ハ）の要素を詳細に説明する。

〔文字群分類部１０〕文字群分類部１０は、図８に示すように、群分類部１１と群確率保持部１２とからなる。

【００５９】群分類部１１は、文字列を入力し、該文字列に含まれる文字（シンボルともいう）を、文字群１、文字群２及び文字群３のいずれかに分類して、分類した文字群の群番号を出力する。群分類部１１は、シンボルと群番号とを対応させて格納する対応表１１ａを有している。この対応表１１ａに格納された群番号は、確率モデル作成部２０及び符号部３０に出力される。

【００６０】群確率保持部１２は、群分類部１１から群番号を入力し、各文字群ごとの出現頻度を出力する。群確率保持部１２は、群番号と群毎の確率とを対応させて格納する対応表１２ａを有している。この対応表１２ａに格納された群出現確率は、符号部３０に出力される。

【００６１】〔確率モデル作成部２０〕確率モデル作成部２０は、辞書２１と、カウンタ２２とからなる。辞書２１は、文字列を入力すると共に、文字群分類部１０より入力された文字が属する群番号を入力して、群番号順位（群中の出現頻度順位）を出力する。そして、辞書２１は、文字群毎に、シンボルと群番号順位とを対応させて格納する対応表２１ａを有している。この対応表２１ａに格納された群番号順位は、符号部３０に出力される。

【００６２】カウンタ２２は、辞書２１から群番号順位を入力し、文字出現確率を出力する。そして、カウンタ２２は、文字群毎に、群中の出現頻度順位と文字出現頻度とを対応させて格納する対応表２２ａを有している。

【００６３】〔符号部３０〕符号部３０は、テーブル３１と、算術符号化部３２とからなる。テーブル３１は、文字群分類部１０から「群番号及び群出現確率」を入力すると共に、確率モデル作成部２０から「群内文字順位及び群内文字出現確率」を入力する。そして、テーブル３１は、群番号と累積出現頻度とを対応させて格納するテーブル３１ａと、文字群毎に、郡内文字順位と累積出現頻度とを対応させて格納する複数のテーブル３１ｂを有している。

【００６４】算術符号化部３２は、テーブル３１が保持する累積出現頻度を入力して、算術符号を出力する。ここで、どの文字がどの群に属するかという情報と文字群の出現頻度に関する情報は、予め予想される出現頻度に従って初期的に与えられる。例えば、図９に示すように、スペース（空白）、Ｅ、Ｔ等の文字は、高出現文字群に分類され、Ｈ、Ｄ、Ｌ等の文字は、低出現文字群に分類される。そして、各々の群出現確率は、それぞれの群に属する文字の個々の出現確率の総和をとったものとする。

【００６５】＜実施例の動作＞次に、図１０を参照して、実施例の動作を説明する。まず、ステップ１００１では、算術符号化する符号語の初期設定として、上端＝１、下端＝０、区間幅＝１．０ とする。

【００６６】ここで、文字群分類部１０は、予め予想される出現頻度に基づいて、群分類部１１の群分類と群確率保持部１２の群確率とを初期化する。なお、群分類の初期化とは、各群の構成要素とどの文字がどの群に属するのかという情報を与えることであり、群確率の初期化とは、例えば、群１：群２：群３＝３：５：１と群の出現確率を初期値に従って与えることである。

【００６７】そして、確率モデル作成部２０は、各シンボル群に分類し、各シンボルごとのカウンタ２２を準備し１に初期化する。また、確率モデル作成部２０は、群累積出現頻度を累積して計算すると共に、各文字群毎に別々の各シンボルの順位、累積出現頻度を計算する。なお、群累積出現頻度を累積して計算するとは、例えば、群３～群Ｍの出現頻度を足し合わせて群２の累積出現頻度とすることをいう。

【００６８】次に、入力文字列より一文字（“ｋ”とする）入力する（ステップ１００２）毎に、文字群分類部１０は、群分類部１１の辞書を検索して入力文字が属する群（“Ｋ”とする）を判別する（ステップ１００３）。

【００６９】ここで、確率モデル作成部２０は、ステップ１００３で判別された群と入力文字に基づいて辞書２１を検索し、出現頻度順位と、群の各文字出現頻度を出力する。

【００７０】そして、算術符号部３０は、文字群累積出現頻度を使用して文字群Ｋを算術符号化する（ステップ１００４）と共に、入力文字ｋを算術符号化する（ステップ１００５）。なお、ステップ１００４の算術符号化は、（イ）群番号及び群累積出現頻度を用いて入力文字群の区間の上端と下端を求めること、（ロ）入力文字の群内出現頻度順位及び当群の累積出現頻度を用いて入力

文字の区間の上端と下端を求めること、（ハ）区間の任意の値を符号として出力すること、により行われる。

【００７１】そして、カウンタ２２にて入力文字ｋの出現頻度を“１”増やし（ステップ１００６）、頻度順に文字群Ｋの辞書を並び替える（ステップ１００７）。次に、“１”増加した文字に伴い、出現頻度順位及び累積出現頻度を更新する（ステップ１００８）。その後、ステップ１００２から実行を繰り返す。

【００７２】〔－１次、０次のBlending方法を用いた算術符号化の動作〕次に、－１次、０次のBlending方法を用いた算術符号化の動作を、図１１を参照して説明する。

【００７３】まず、ステップ１１０１では、算術符号化の初期設定として、（イ）文字群累積出現頻度を準備すること、（ロ）各文字出現頻度を０とすること、（ハ）未出現文字として各文字群毎に全文字を登録すること、（ニ）各文字群毎に準備した未出現文字確率を１に設定すること、を行う。

【００７４】次に、入力文字列より一文字（“ｋ”とする）入力する（ステップ１１０２）毎に、文字群分類部１０は、群分類部１１の辞書を検索して入力文字が属する群（“Ｋ”とする）を判別する（ステップ１１０３）。

【００７５】そして、算術符号化部３０は、文字群累積出現頻度を使用して、文字群Ｋを算術符号化する（ステップ１１０４）。ここで、文字群Ｋが先に出現していたか否かが判断される（ステップ１１０５）。　ステップ１１０５で、先に出現していたと判断された場合、文字群Ｋの累積出現頻度を使用して、文字ｋを算術符号化し（ステップ１１０６）、文字ｋをカウントする（ステップ１１０７）とともに、頻度順に辞書を並び替える（ステップ１１０８）。

【００７６】一方、ステップ１１０５で、先に出現していないと判断された場合、文字群Ｋの未出現文字区間を算術符号化し（ステップ１１０９）、文字ｋを算術符号化する（ステップ１１１０）とともに、文字ｋを文字群Ｋの辞書に挿入し、文字ｋを文字群Ｋの未出現文字より除く（ステップ１１１１）。なお、ステップ１１１０では、文字群Ｋの全未出現文字は、等確率にされる。

【００７７】ステップ１１０８とステップ１１１１の後に、文字群Ｋの累積出現頻度が更新される。

〔符号部３０の算術符号化の具体例〕図１２は、符号部３０の算術符号化の具体例を示す図である。図１２では、「ひらがな」を文字群１、「スペース、句読点、改行マーク」を文字群２、その他の「数字」等を文字群３としている。一文字群としている。「ひらがな」の出現確率は０．５２で、「スペース、句読点、改行マーク」の出現確率は０．１３である。圧縮の初期では、どの文字も出現したことがなく、各文字の出現頻度は０である。

【００７８】この場合、従来の方式では、どの文字も等確率で出現可能と考えて、等幅の符号区間を設定するが、本実施例では、図１２（Ｂ）に示すように各群の出現確率に応じて定め、文字群区間の中でその文字群に属する各文字を等幅とする。各文字群区間は、前述した各群出現確率（図１２（Ａ）参照）に従って分ける。

【００７９】本発明の文字群区間を定めた上で各文字区間を定める方式によると、図１２（Ｂ）に示すように、圧縮初期の段階から出現確率の高い文字に対して広い符号区間を与えることができる。

【００８０】＜本実施例の変形例＞前記実施例では、文字群出現確率を固定的なものとしてきたが、文字群出現確率を動的に変える変形例を述べる。

【００８１】（１）文字群出現確率を個々の群の出現確率を動的に変えるもの

（２）群の文脈を取り入れた、条件付出現確率を動的に変えるもの

まず、文字群出現確率を、個々の群の出現確率を動的に変えるものを図１３に示す。これは、図７における文字群分類部１０に相当する。文字群分類部１０は、どの文字がどの群に属するかを示す群分類部１１と、各文字群に初期値を与え、文字を入力するごとに、その文字の属する群の出現頻度を“１”増やし、群累積出現頻度を更新する群カウンタ１３とから構成されている。

【００８２】その動作は、図１４を参照して説明する。まず、ステップ１４０１では、初期設定として、文字群累積出現頻度をとって、各文字出現頻度を１とし、各文字群毎に累積出現頻度をとる。

【００８３】次に、入力文字列より一文字（“ｋ”）入力する（ステップ１４０２）毎に、文字群分類部１０は、群分類部１１の辞書を検索して入力文字が属する群（“Ｋ”とする）を判別する（ステップ１４０３）。

【００８４】そして、算術符号部３０は、文字群累積出現頻度を使用して文字群Ｋを算術符号化する（ステップ１４０４）と共に、入力文字ｋを算術符号化する（ステップ１４０５）。

【００８５】そして、文字ｋの出現頻度と文字群Ｋの出現頻度を、それぞれ１ずつ増加させ（ステップ１４０６）、頻度順に文字群Ｋの辞書を並び替える（ステップ１４０７）。

【００８６】次に、ステップ１４０６の増加に伴い、文字群Ｋの累積出現頻度を更新する（ステップ１４０８）。同様に、群の文脈を取り入れ、条件付出現確率を動的に得ることもできる。０次の値は初期値を与え、１次以上の条件付確率は、図５に示すように、各ノード群を通る文字群が出る毎に、出現回数を各ノードにて計数しておくことによって条件付確率が求められる。ここで従来は、シンボルが木の各節点になっていたのに対し、本実施例では、群が木の各節点になっている。

【００８７】群出現頻度に１次の条件付出現確率をとる

場合のフローを図１５に示す。まず、初期化として以下の（イ）～（ヘ）を行う（ステップ１６０１）。
（イ）各文字群出現頻度を初期化する。
（ロ）文字群累積出現頻度をとる。
（ハ）各文字出現頻度を“１”とする。
（ニ）各文字群毎に累積出現頻度をとる。
（ホ）一つ前の文字の群番号を保持する。
（ヘ）一つ前の群番号を保持しておくレジスタＲ（＝文脈）を初期化する。

【００８８】次に、一文字（ｋとする）を入力する（ステップ１６０２）。そして、どの文字群（Ｋとする）に入力文字ｋが属するかを判別する（ステップ１６０３）。

【００８９】そして、「RKの出現頻度／Rの出現頻度」を意味する条件付き確率P(K¦R)を符号部３０にて算術符号化する。つまり、R に続いてそれぞれの群が起こる確率に従って区間を分割し、このうち群K の区間を選択する。なお、各群の区間は、Rに続いて起こる文字群の累積出現頻度によってその下限が求まる（ステップ１６０４）。

【００９０】そして、条件付き確率P(k¦K)を算術符号化すると共に、文字群の条件付累積出現頻度CF(k¦K)を使用して、入力文字ｋを算術符号化する（ステップ１６０５）。

【００９１】そして、文字出現頻度C(k¦K)、C(K¦R)の値をそれぞれ“１”だけ増加させる（ステップ１６０６）。そして、文字群Ｋの辞書を文字出現頻度C(x¦K)に従って並び替える（ステップ１６０７）。

【００９２】そして、文字群Ｋの文字累積出現頻度CF(x¦K)を更新すると共に、文字群Ｒに続く文字群の群累積出現頻度CF(X¦R)を更新する（ステップ１６０８）。そして、レジスタＲに入力文字ｋを設定する（ステップ１６０９）。

【００９３】以後、ステップ１６０２からの処理を繰り返す。

＜実施例の効果＞次に、実施例のデータ圧縮効果を図１６を参照して説明する。

【００９４】図１６（Ａ）は、データ圧縮率が圧縮対象ファイルのサイズによってどう変化するかを、本実施例による場合、静的符号化方式（準適応型）による場合及び適応型符号化方式による場合の３つのケースについて示したものである。図１６（Ａ）の横軸と横軸は、それぞれ圧縮対象ファイルサイズとデータ圧縮率であり、線７ａは、本実施例による場合、線７ｂは、静的符号化方式による場合、線７ｃは、適応型符号化方式による場合をそれぞれ示している。

【００９５】図１６（Ａ）から明らかなように、静的符号化方式による場合は、圧縮対象ファイルのサイズにかかわらずほぼ一定のデータ圧縮率を保持し、比較対象の中では最もデータ圧縮できていることが分かる。一方、適応型符号化方式と本実施例による場合は、圧縮対象ファイルのサイズが大きくなるほど圧縮率が小さく、即ち、よりよくデータ圧縮され、静的符号化方式のデータ圧縮率に近づくことが分かる。そして、本実施例による場合は、適応型符号化方式による場合よりも常にデータ圧縮率が小さくなっている。

【００９６】ここで、圧縮対象ファイルサイズがほぼ０の時における、適応型符号化方式による場合と静的符号化方式による場合のデータ圧縮率の差は、静的符号化方式には、各文字出現頻度の初期値が与えられているためである。

【００９７】また、圧縮対象ファイルサイズがほぼ０の時における、適応型符号化方式による場合と本実施例による場合とのデータ圧縮率の差は、本実施例には、各群出現頻度の初期値が与えられているためである。

【００９８】次に、図１６（Ｂ）は、圧縮前のファイルサイズによって圧縮後のファイルサイズがどう変化するかを、本実施例による場合、静的符号化方式（準適応型）による場合及び適応型符号化方式による場合の３つのケースについて比較したものである。なお、図１６（Ｂ）中には、符号化を行わない場合も参考に記してある。図１６（Ｂ）の横軸と縦軸は、それぞれ圧縮前のファイルサイズと圧縮後のファイルサイズであり、線７ｄは、本実施例による場合、線７ｅは、静的符号化方式による場合、線７ｆは、適応型符号化方式による場合、線７ｇは、符号化を行わない場合をそれぞれ示している。

【００９９】図１６（Ｂ）から明らかなように、符号化を行ういずれの場合にも、圧縮前のファイルサイズが大きくなるほど、圧縮後のファイルサイズの増加は鈍ることが分かる。また、静的符号化方式による場合と本実施例による場合は、ファイルサイズが小さい場合に、圧縮後のファイルサイズが圧縮前のファイルサイズよりも大きくなることが分かる。

【０１００】そして、所定値よりも圧縮前のファイルサイズが小さいときは、静的符号化方式による場合、本実施例による場合、適応型符号化方式による場合の順に圧縮後のファイルサイズが小さくなるが、前記所定値よりも圧縮前のファイルファイルサイズが大きくなると、適応型符号化方式による場合、静的符号化方式による場合、本実施例による場合の順に圧縮後のファイルサイズが小さくなることが分かる。

【０１０１】ここで、圧縮前のファイルサイズがほぼ０の時における、静的符号化方式に補助データを付加した場合のファイルサイズが０でないのは、各文字出現頻度の初期値情報を補助データとして持つためである。

【０１０２】また、圧縮前のファイルサイズがほぼ０の時における、本実施例に補助データを付加した場合のファイルサイズが０でないのは、各群出現頻度の初期値情報を補助データとして持つためである。

【０１０３】

【発明の効果】本発明の第１のデータ圧縮方法及びデータ圧縮装置によれば、文字を互いに同じ統計的性質を有する文字ごとに複数の群に分類し、さらに、それぞれの群の出現確率を計算するようにしたため、従来の方法に比べ、初期の段階で最適な符号領域を割り振ることができる。これは、出現可能な文字数が多く、圧縮対象ファイルが小さいときに特に有効である。つまり、従来の適応型符号化方式では、確率モデルを構築するのにある程度の長さの入力列を必要とし、圧縮対象のサイズが小さい場合は圧縮率が上がらないのに対して、本発明では、十分な圧縮率を得ることができる。

【０１０４】本発明の第２及び第３のデータ圧縮方法によれば、第１のデータ圧縮方法に比べ、データの出現確率に従った符号を予め割り当てることができるため、ファイルサイズが小さい場合にも高い圧縮率を得ることが可能になる。

【０１０５】本発明の第４のデータ圧縮方法によれば、入力データに従って出現頻度を計算し直すので、徐々にデータに即した出現頻度に基づく圧縮が可能になる。本発明の第５及び第６のデータ圧縮方法によれば、直前に出現した文字が属する群あるいは直々前に出現した文字が属する群を条件とした条件付確率を用いることで、さらに高い圧縮率が得られるようになる。

【図面の簡単な説明】

【図１】本発明のデータ圧縮の原理図である。（Ａ）はデータ圧縮方法の原理図を示し、（Ｂ）はデータ圧縮装置の原理図を示す。

【図２】多値算術符号の原理を示す図である。（Ａ）は、各文字の出現頻度を示している。（Ｂ）は、出現頻度順の累積出現頻度を示している。（Ｃ）は、算術符号化の原理を示している。

【図３】算術符号化の装置構成を示す図である。

【図４】従来の多値算術符号化のフローを示す図である。

【図５】文脈の木構造（２次の場合）を示す図である。

【図６】従来の多値算術符号化（－１、０次のブレンディング）のフローを示す図である。

【図７】実施例の装置構成の概略を示す図である。

【図８】実施例の装置構成を詳細に示す図である。

【図９】群分類と群出現確率を示す図である。

【図１０】実施例の多値算術符号化のフローを示す図である（その１）。

【図１１】実施例の多値算術符号化のフローを示す図である（その２）。このフローは、－１、０次のブレンディングになっている。

【図１２】文字群出現確率及び初期符号区間を示す図である。（Ａ）は、文字群出現確率を示している。（Ｂ）は、確率モデル未保持区間時点における符号区間を示している。

【図１３】文字群分類部を示す図である。

【図１４】実施例の多値算術符号化のフローを示す図である（その３）。

【図１５】実施例の多値算術符号化のフローを示す図である（その４）。

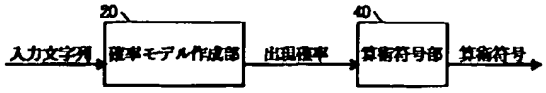
【図１６】従来の算術符号化と本実施例の算術符号化との効果の比較図である。（Ａ）は、圧縮対象ファイルサイズが変化した場合におけるデータ圧縮率の変化を示している。（Ｂ）は、圧縮前のファイルサイズが変化した場合における圧縮後のファイルサイズの変化を示している。

【符号の説明】

Ｓ１　群構成ステップ
Ｓ２　群出現確率計算ステップ
Ｓ３　群中文字出現確率計算ステップ
Ｍ１　群構成部
Ｍ２　群出現確率計算部
Ｍ３　群中文字出現確率計算部
１０　文字群分類部
１１　群分類部
１２　群確率保持部
１３　群カウンタ
２０　確率モデル作成部
２１　辞書
２２　カウンタ
３０　符号部
３１　テーブル
３２　算術符号化部
４０　算術符号部

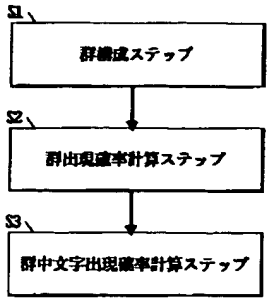
【図３】

算術符号化の装置構成を示す図



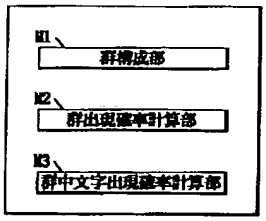
【図1】

本発明によるデータ圧縮方法の原理図



（A）

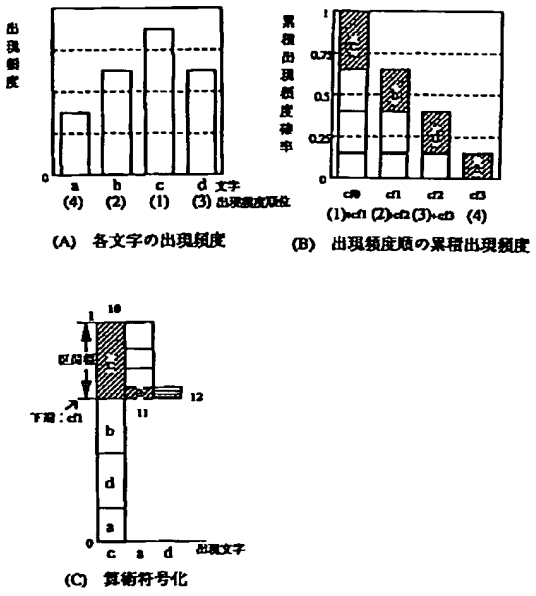
本発明によるデータ圧縮装置の原理図



（B）

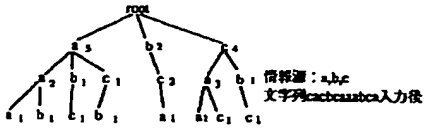
【図2】

多値算術符号の原理を示す図



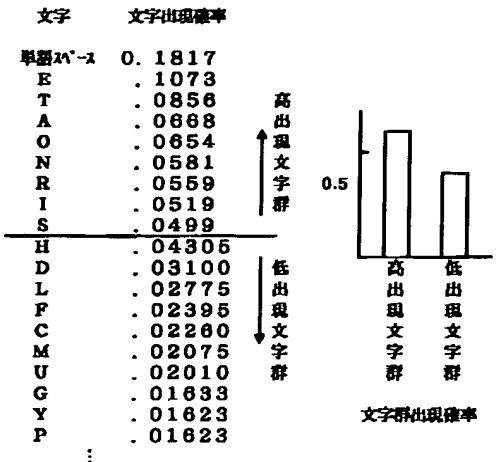
【図5】

文脈の木構造（２次の場合）を示す図



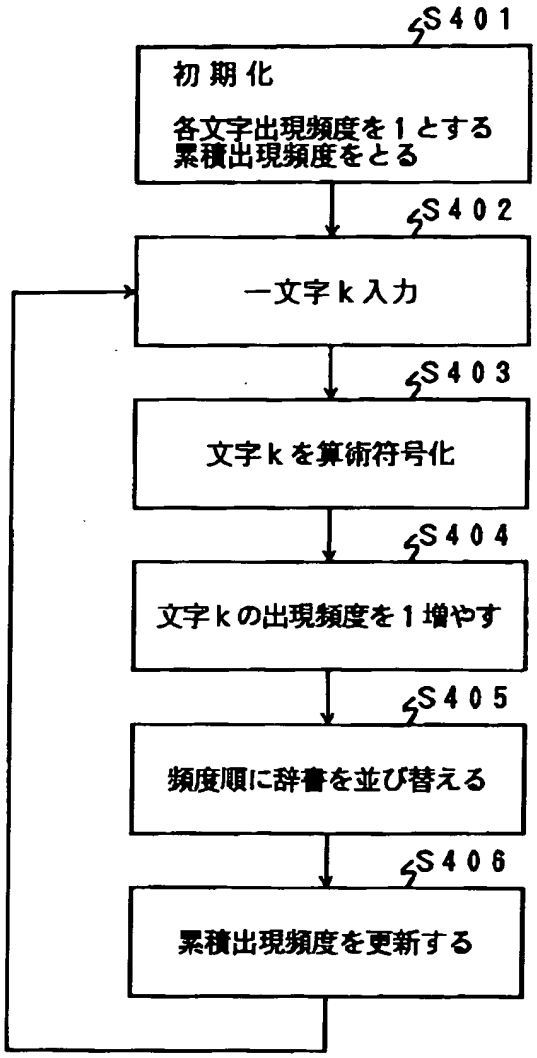
【図9】

群分類と群出現確率を示す図



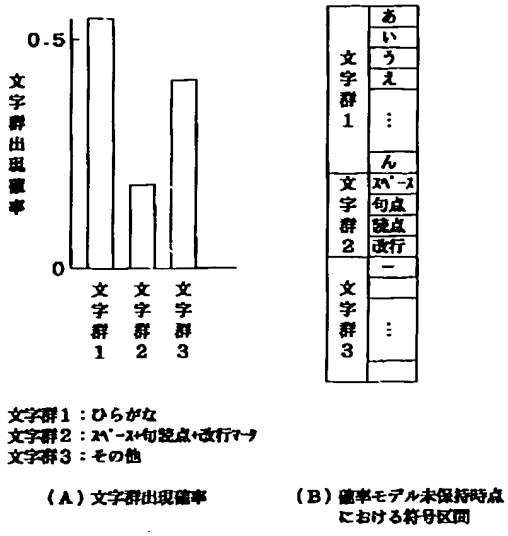
【図4】

従来の多値算術符号化のフローを示す図



【図12】

文字群出現確率及び初期符号区間を示す図

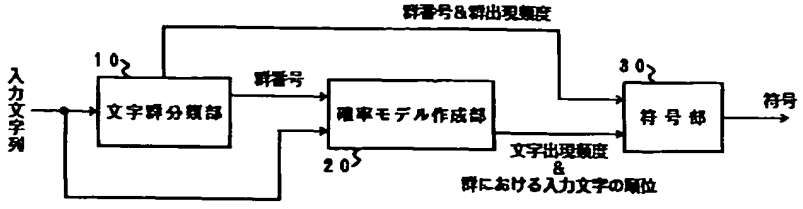


文字群1：ひらがな
文字群2：ｽﾍﾟｰｽ+句読点+改行ﾏｰｸ
文字群3：その他

（A）文字群出現確率　　（B）確率モデル未保持時点における符号区間

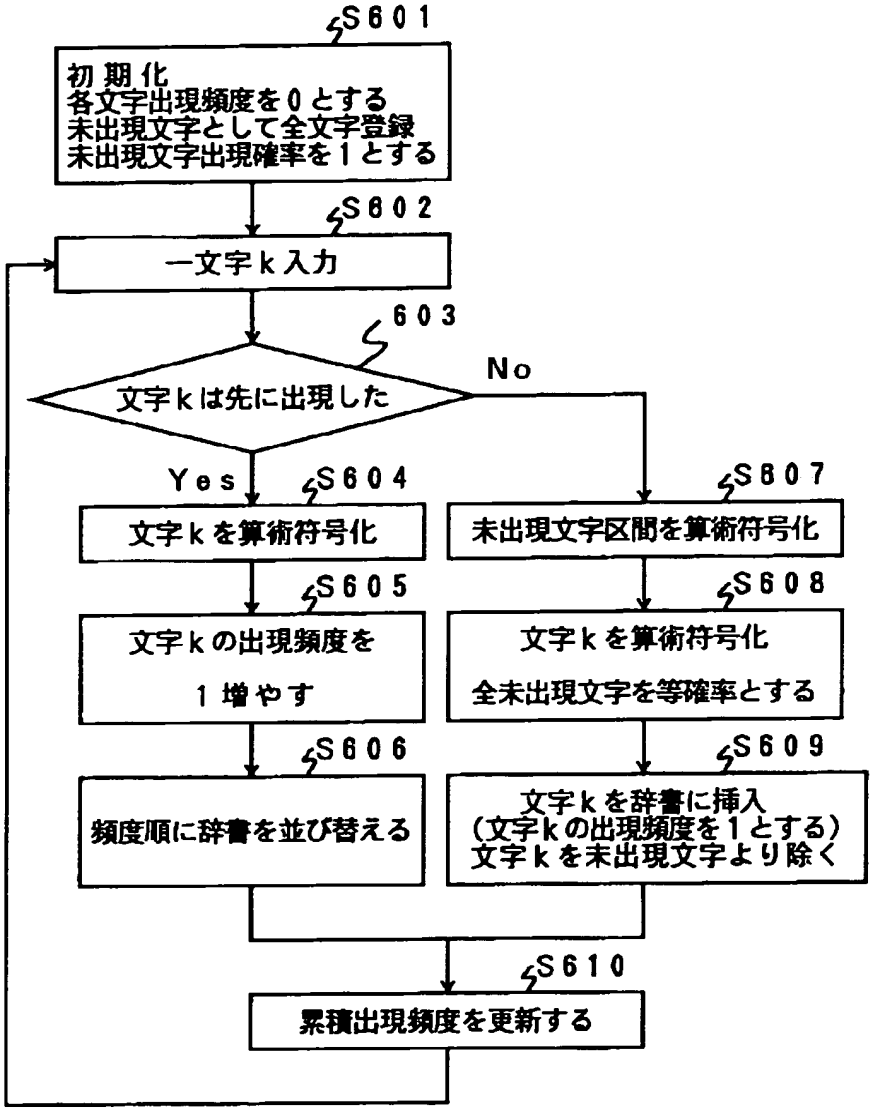
【図7】

実施例の装置構成の概略を示す図



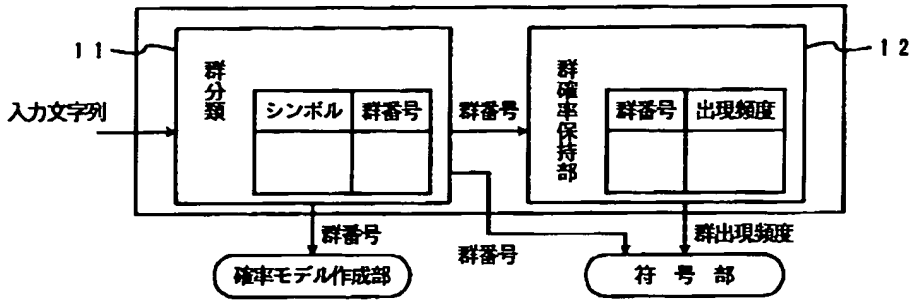
【図6】

従来の多値算術符号化（－1，0次のブレンディング）のフローを示す図

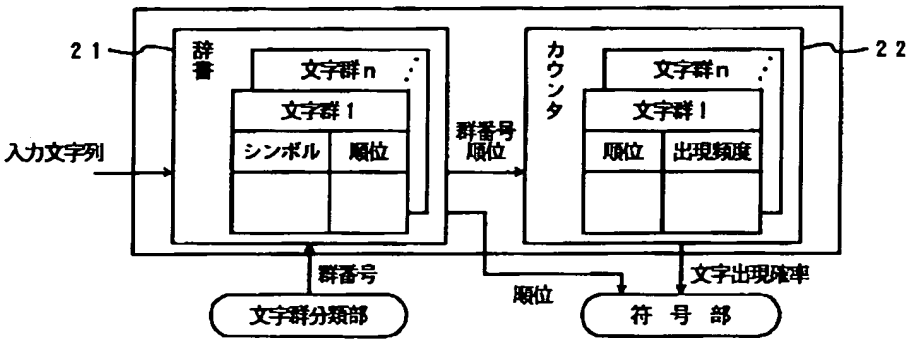


【図8】

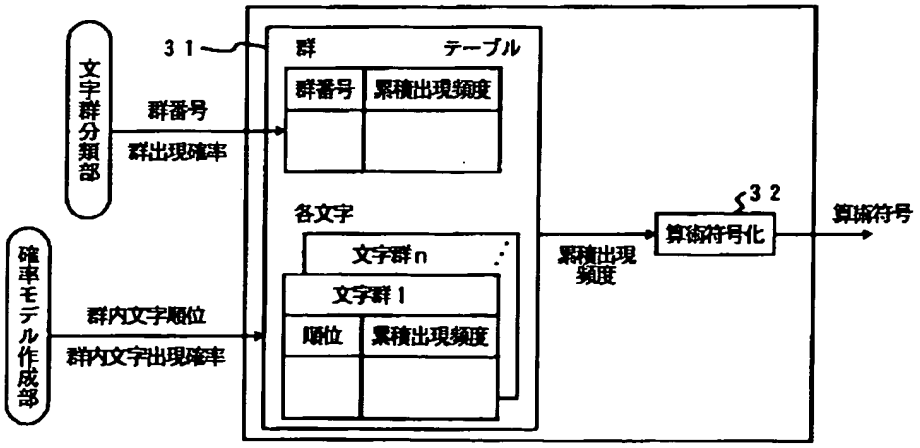
実施例の装置構成を詳細に示す図



(A) 文字群分類部



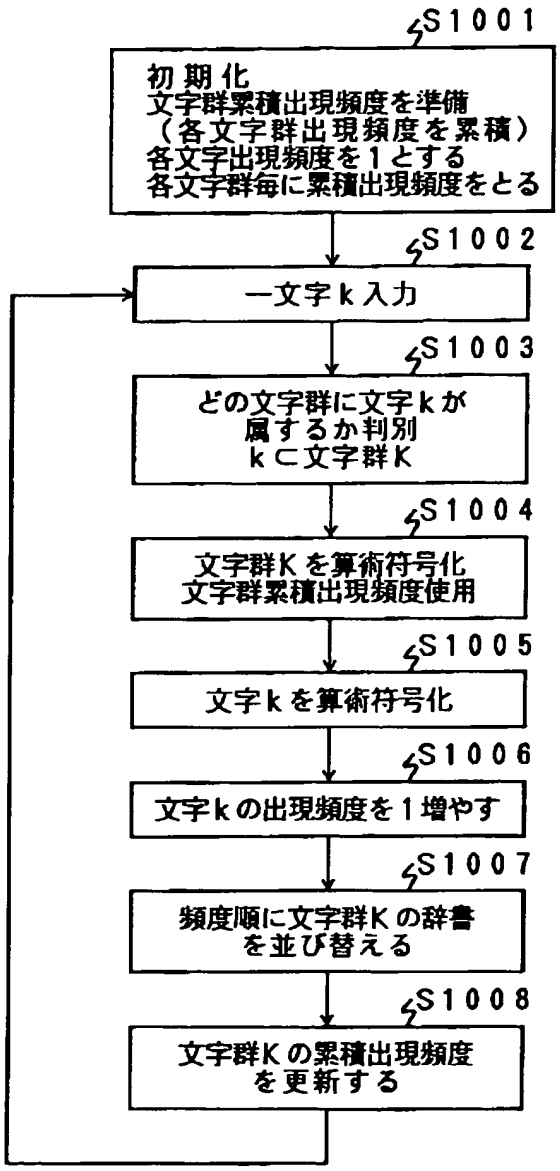
(B) 確率モデル作成部



(C) 符号部

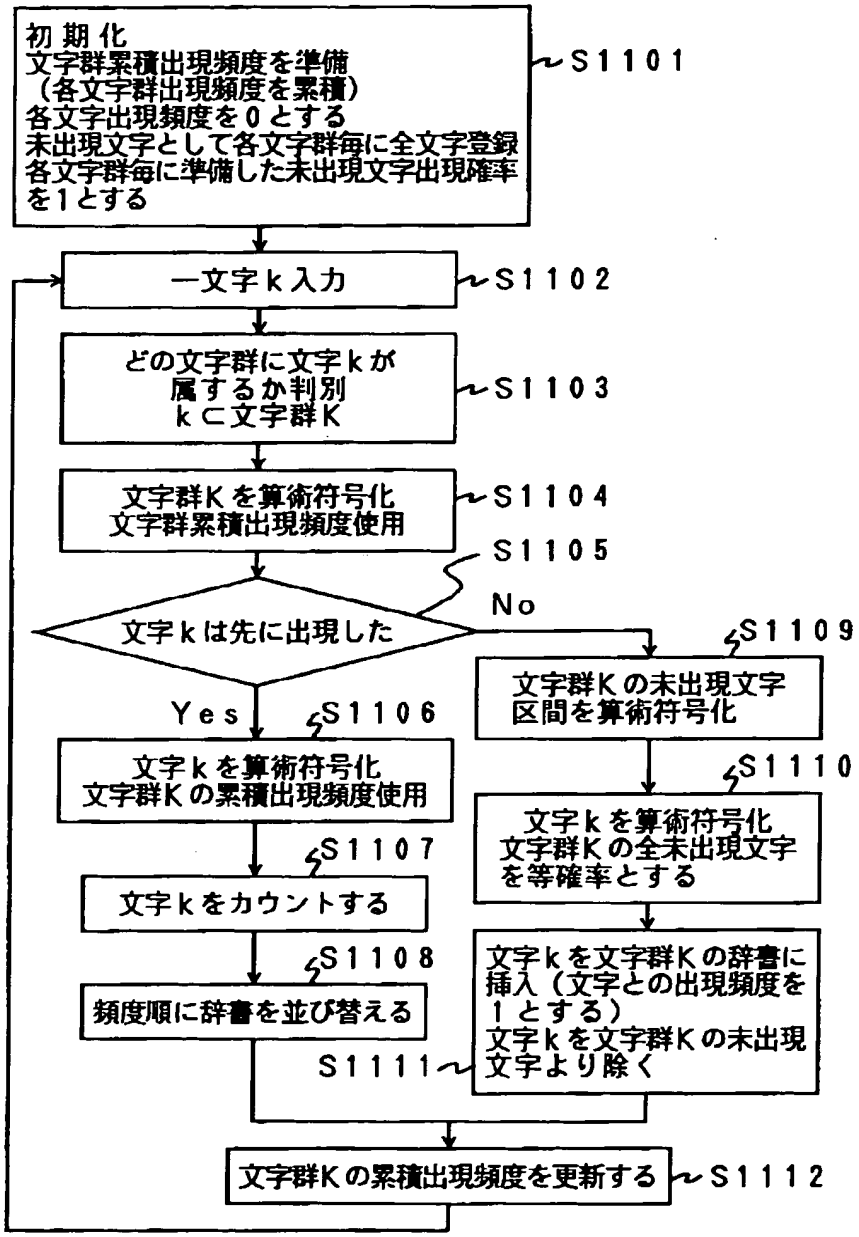
【図10】

実施例の多値算術符号化のフローを示す図（その1）



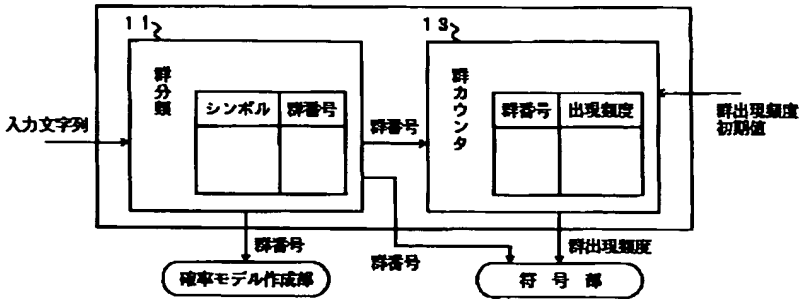
【図11】

実施例の多値算術符号化のフローを示す図（その2）



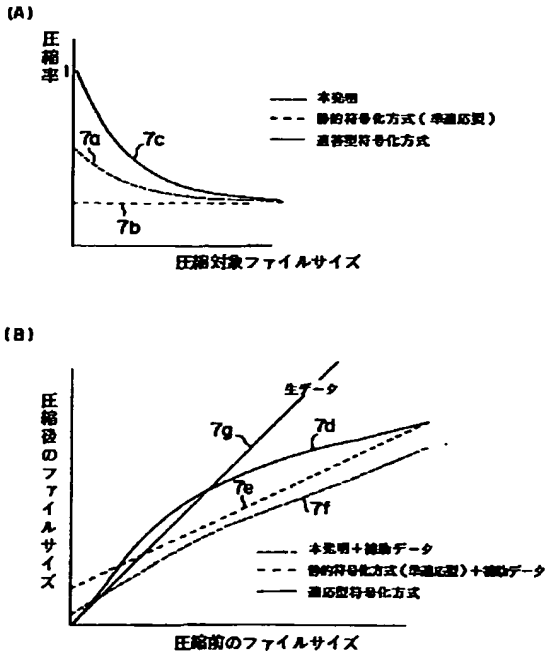
【図13】

文字群分類部を示す図



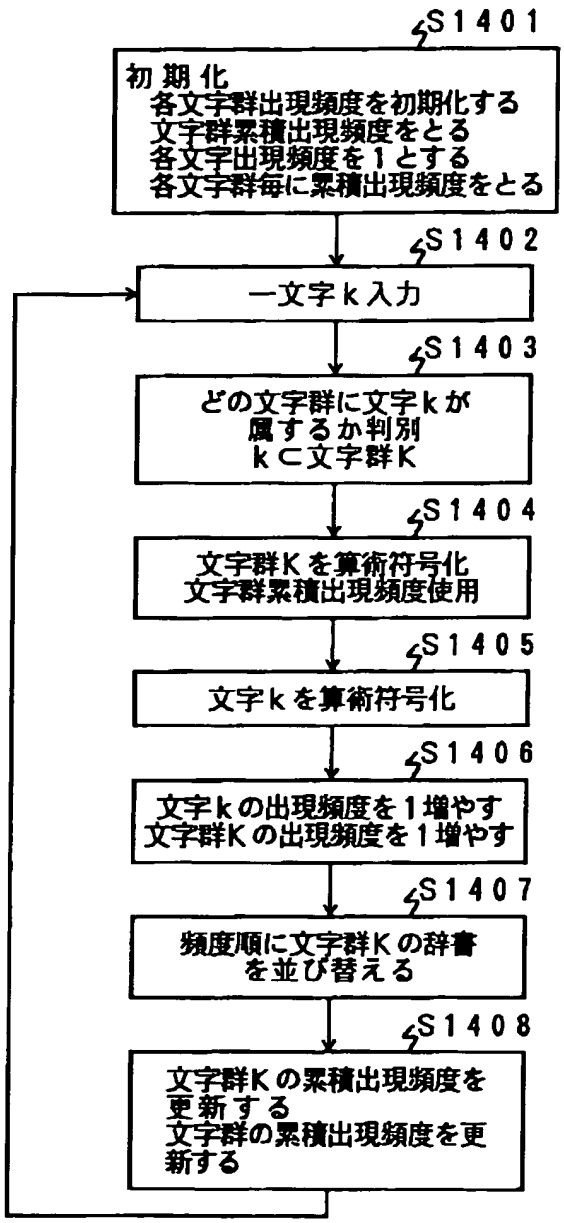
【図16】

従来の算術符号化と本実施例の算術符号化との効果の比較図



【図14】

実施例の多値算術符号化のフローを示す図（その3）



【図15】

実施例の多値算術符号化のフローを示す図